A Project Report

On

PREDICTIVE MODELING OF CONGENITAL HEART DISEASE USING MACHINE LEARNING TECHNIQUES

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

In Partial Fulfillment of the Requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted By

M. DIVIJA 21691A0540

Under the guidance of

Mr.Ajaypradeep Natarajsivam M.E.,(Ph.D.)

Assistant Professor

Department of Computer Science & Engineering

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)

Accredited by NBA, Approved by AICTE, New Delhi)

AN ISO 21001:2018 Certified Institution

P. B. No: 14, Angallu, Madanapalle – 517325

2024-2025

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC – AUTONOMOUS)
Approved by AICTE, New Delhi and Affiliated to JNTUA, Ananthapuramu
www.mits.ac.in www.mits.edu

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **“PREDICTIVE MODELING OF CONGENITAL HEART DISEASE USING MACHINE LEARNING TECHNIQUES”** is a bonafide work carried out by

M. DIVIJA - (21691A0540)

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Computer Science & Engineering** in **Madanapalle Institute of Technology & Science, Madanapalle**, affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2024-2025.

Guide	Head of the Department
Mr. Ajaypradeep Natarajsivam M.E.,(Ph.D.)	**Dr. M.Sreedevi, Ph.D**
Professor and Head,	Professor and Head,
Department of CSE.	Department of CSE.

Submitted for the University examination held on:

Internal Examiner	**External Examiner**
Date:	**Date:**

AKNOWLEDGEMENTS

I sincerely thank the **Management** of **Madanapalle Institute of Technology & Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

I sincerely thank **Dr.C.Yuvaraj, M.E., Ph.D., Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology & Science,** Madanapalle.

I express my deep sense of gratitude to **Dr. M. Sreedevi, Ph.D., Professor and Head of the Department of CSE** for her continuous support in making necessary arrangements for the successful completion of the Project.

I express my deep sense gratitude to **Mr. Ch. Hemanand, M.Tech., (Ph.D.), Internship Coordinator** for his guidance and encouragement that helped us to complete this project.

I express my deep sense gratitude to **Mrs. M.Bommy, M.E., (Ph.D.), Project Coordinator, and Mr. K. Sathish, M.Tech., (Ph.D.), Project Co Coordinator, Department of CSE** for their guidance and encouragement that helped us to complete this project.

I express my deep gratitude to my guide **Mr. Ajaypradeep Natarajsivam M.E.,(Ph.D.), Assistant Professor Department of CSE** for her guidance and encouragement that helped us to complete this project.

I also wish to place on record my gratefulness to other **Faculty of CSE Department** and also to our friends and our parents for their help and cooperation during our project work.

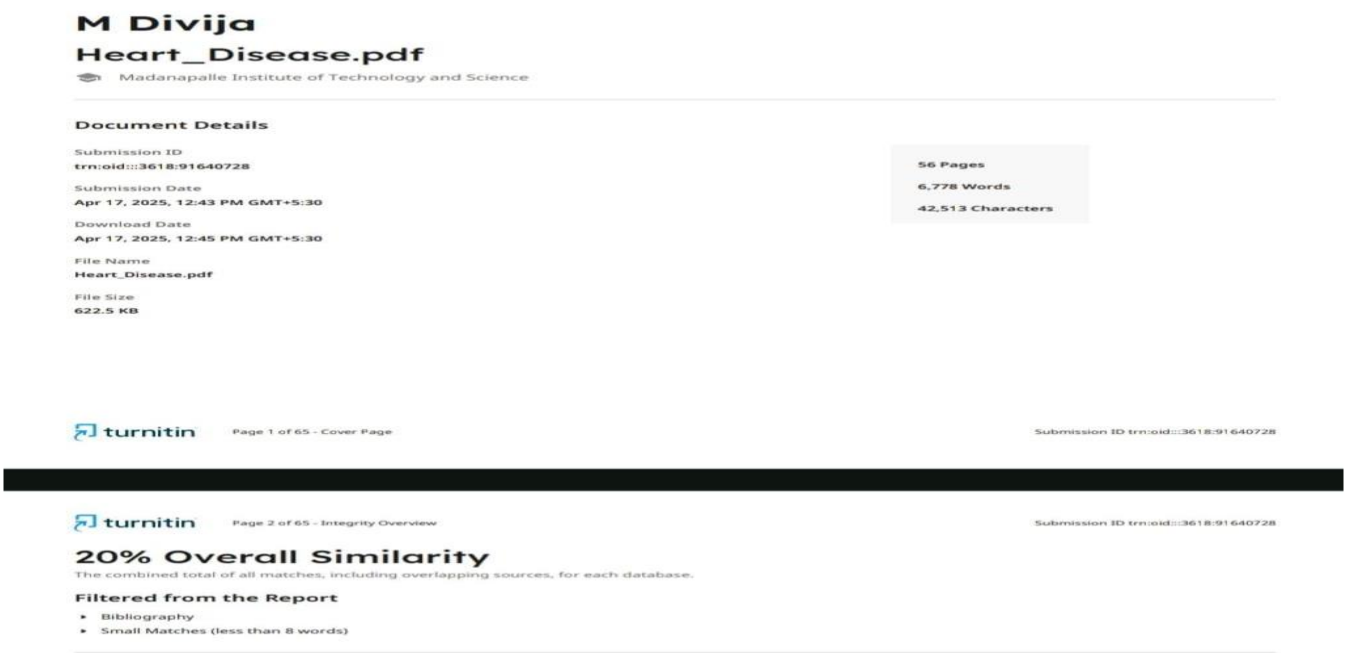
MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC-AUTONOMOUS INSTITUTION)
Affiliated to JNTUA, Ananthapuramu & Approved by AICTE, New Delhi
NAAC Accredited with A+ Grade, NIRF India Rankings 2024 - Band: 201-300 (Engg.)
NBA Accredited - B.Tech. (CIVIL, CSE, CST, ECE, EEE, MECH), MBA & MCA

RECOGNISED RESEARCH CENTER

Plagiarism Verification Certificate

This is to certify that the B. Tech Project report titled, "**PREDICTIVE MODELING OF CONGENITAL HEART DISEASE USING MACHINE LEARNING TECHNIQUES**" submitted by **M.DIVIJA (21691A0540)** has been evaluated using Anti-Plagiarism Software, TURNITIN and based on the analysis report generated by the software, the report's similarity index is found to be 20%

The following is the TURNITIN report for the project report consisting 78 pages



M Divija
Heart_Disease.pdf
Madanapalle Institute of Technology and Science

Document Details

Submission ID
trn:oid:::3618:91640728

Submission Date
Apr 17, 2025, 12:43 PM GMT+5:30

Download Date
Apr 17, 2025, 12:45 PM GMT+5:30

File Name
Heart_Disease.pdf

File Size
622.5 KB

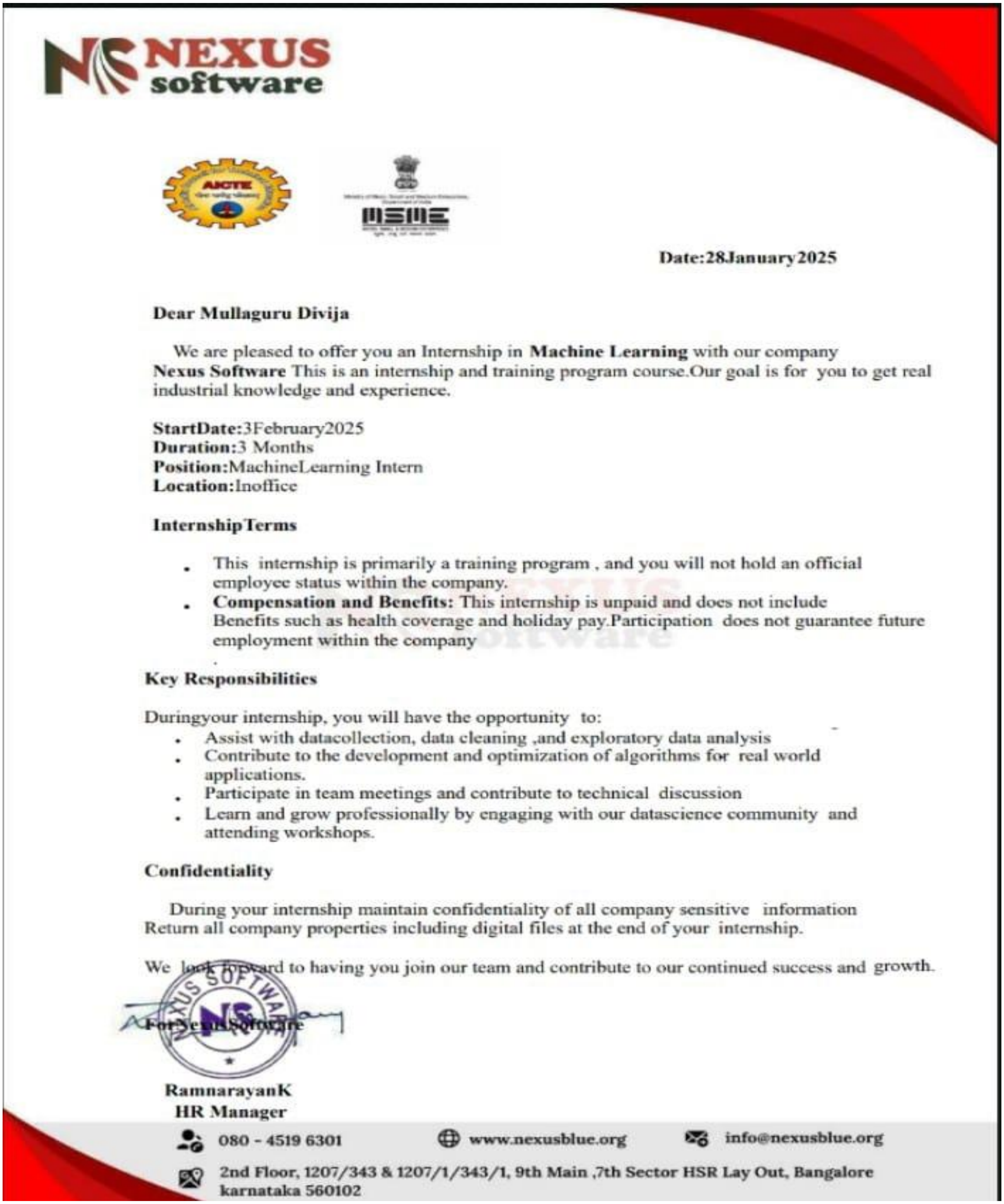
56 Pages
6,778 Words
42,513 Characters

turnitin Page 1 of 65 - Cover Page Submission ID trn:oid:::3618:91640728

turnitin Page 2 of 65 - Integrity Overview Submission ID trn:oid:::3618:91640728

20% Overall Similarity
The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report
- Bibliography
- Small Matches (less than 8 words)



NEXUS software

Date:28January2025

Dear Mullaguru Divija

We are pleased to offer you an Internship in **Machine Learning** with our company **Nexus Software** This is an internship and training program course.Our goal is for you to get real industrial knowledge and experience.

StartDate:3February2025
Duration:3 Months
Position: MachineLearning Intern
Location: Inoffice

InternshipTerms

- This internship is primarily a training program , and you will not hold an official employee status within the company.
- **Compensation and Benefits:** This internship is unpaid and does not include Benefits such as health coverage and holiday pay.Participation does not guarantee future employment within the company

Key Responsibilities

Duringyour internship, you will have the opportunity to:

- Assist with datacollection, data cleaning ,and exploratory data analysis
- Contribute to the development and optimization of algorithms for real world applications.
- Participate in team meetings and contribute to technical discussion
- Learn and grow professionally by engaging with our datascience community and attending workshops.

Confidentiality

During your internship maintain confidentiality of all company sensitive information Return all company properties including digital files at the end of your internship.

We look forward to having you join our team and contribute to our continued success and growth.

ForNexusSoftware

RamnarayanK
HR Manager

080 - 4519 6301 www.nexusblue.org info@nexusblue.org

2nd Floor, 1207/343 & 1207/1/343/1, 9th Main ,7th Sector HSR Lay Out, Bangalore
karnataka 560102

DECLARATION

I hereby declare that the results embodied in this project "**PREDICTIVE MODELING OF CONGENITAL HEART DISEASE USING MACHINE LEARNING TECHNIQUES**" by me under the guidance of **Mr. Ajaypradeep Natarajsivam M.E., (Ph. D)., Assistant Professor, Department of CSE** in partial fulfillment of the award of **Bachelor of Technology in Computer Science & Engineering** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu**. I have not submitted to any other University/Institute for award of any other degree.

Date :

Place :

PROJECT ASSOCIATES
M. DIVIJA

I certify that above statement made by the students is correct to the best of my knowledge.

Date: **Guide:**

TABLE OF CONTENTS

ABSTRACT

Congenital heart disease (CHD) refers to a range of structural abnormalities in the heart that are present at birth. It is among the most common congenital anomalies globally, affecting nearly 1% of live births. Despite significant advancements in neonatal care and surgical interventions, CHD remains a leading cause of infant morbidity and mortality. Early and accurate diagnosis is crucial to ensure timely medical or surgical management, especially during the critical early stages of development. However, the diagnosis of CHD can be particularly challenging in resource-constrained environments due to the requirement for specialized equipment and trained pediatric cardiologists. Echocardiography remains the standard imaging technique for diagnosing CHD due to its non-invasive nature and real-time imaging capabilities. Nonetheless, accurate interpretation of echocardiographic images requires considerable expertise. To overcome this barrier and improve diagnostic consistency, researchers have increasingly turned to machine learning (ML) and deep learning (DL) methods. These technologies offer the potential to automatically detect and classify CHD patterns from medical imaging data such as echocardiograms and cardiac MRIs. Deep learning architectures like Convolutional Neural Networks (CNNs), U-Net, and DenseNet have shown promising results in CHD applications. U-Net is particularly effective in segmenting anatomical structures of the heart, while DenseNet provides robust performance in classifying various CHD types due to its dense connectivity and feature reuse capabilities. These models help identify septal defects, chamber abnormalities, and other anomalies with high accuracy.

.

LIST OF FIGURES

ABBREVIATIONS

CNN	-	Convolutional Neural Network
CHD	-	Congenital Heart Disease
DL	-	Deep Learning
ML	-	Machine Learning
AI	-	Artificial Intelligence

CHAPTER-1

INTRODUCTION

1.1 Introduction

Congenital heart disease (CHD**)** remains a leading cause of infant mortality globally, despite improvements in surgical treatments and neonatal care. One of the primary challenges in managing CHD is timely and accurate diagnosis, especially in under-resourced healthcare systems where access to pediatric cardiologists and advanced imaging tools is limited. Since CHD often involves complex structural anomalies, early detection is critical for improving patient outcomes and guiding surgical planning during early developmental stages.

Traditional diagnostic approaches rely heavily on echocardiography, which—though effective and non-invasive—requires skilled interpretation, making it susceptible to diagnostic variability. In response, the use of automated diagnostic systems powered by deep learning has gained momentum in recent years. Deep learning models, especially Convolutional Neural Networks (CNNs), have proven capable of interpreting complex echocardiographic and MRI data, significantly reducing the diagnostic burden on clinicians and minimizing human error.

This study proposes a deep learning-driven framework for the early detection and classification of CHD, focusing on improving diagnostic consistency and accessibility in low-resource settings. Advanced architectures such as U-Net, DenseNet, and ResNet have been explored for tasks like heart chamber segmentation, septal defect identification, and structural anomaly classification. U-Net, in particular, has demonstrated high performance in segmenting cardiac structures, while DenseNet's feature reuse ability enhances classification accuracy.

Given the scarcity of large, annotated pediatric datasets and the variability in heart morphology among children, the system incorporates data augmentation and transfer learning techniques to improve model robustness. Among the tested models, DenseNet emerged as the most balanced in terms of accuracy and computational efficiency, making it suitable for real-world application in pediatric cardiology.

1.2 Motivation

The critical need for early and precise diagnosis of congenital heart disease (CHD)—a leading cause of neonatal and infant mortality—drives the motivation behind this initiative. CHD encompasses a spectrum of structural abnormalities present at birth, which, if undetected in time, can result in life-threatening complications. Early diagnosis is often hindered by limited access to pediatric cardiologists and reliance on manual interpretation of echocardiographic data, especially in under-resourced settings. These challenges delay intervention, adversely impacting survival rates and long-term health outcomes.

This study aims to address these diagnostic challenges by leveraging the power of deep learning to assist in the detection and classification of CHD using medical imaging data. Manual assessments, while standard, are prone to variability and require high levels of expertise. Advanced neural network architectures such as U-Net, DenseNet, and ResNet offer a promising alternative by learning complex features from echocardiograms and MRI scans—features that may be too subtle for human detection.

By developing an intelligent, cost-effective, and scalable diagnostic framework, the project seeks to empower healthcare workers in diverse environments. The deep learning models integrated into this system are optimized not only for high accuracy but also for efficient deployment on edge devices, ensuring accessibility in settings with limited computational resources.

Beyond its immediate clinical utility, this initiative also contributes to a larger vision of integrating AI-driven diagnostic tools into pediatric cardiology. It fosters interdisciplinary collaboration between clinicians, data scientists, and technologists to build adaptive systems that evolve with medical advancements. The continuous learning capacity of these models supports long-term relevance, making them integral to the future of precision pediatric care. Ultimately, this work lays the groundwork for a healthcare ecosystem where early CHD diagnosis becomes universally accessible and consistently accurate, regardless of geography or resource availability.

1.3 Problem Definition

Congenital Heart Disease (CHD) is among the most prevalent and life-threatening congenital conditions, affecting approximately 1% of live births worldwide. Despite notable progress in neonatal care and pediatric cardiac surgery, CHD remains a significant cause of infant morbidity and mortality. One of the most persistent challenges in combating CHD lies in early and accurate diagnosis, which is crucial for initiating timely medical or surgical interventions. Unfortunately, healthcare disparities—particularly in low-resource settings—limit access to specialized imaging tools and trained pediatric cardiologists, leading to delayed or inaccurate diagnoses.

CHD encompasses a broad spectrum of structural abnormalities in the heart, including septal defects, valve malformations, and outflow tract obstructions. These anomalies are complex and often subtle in early stages, making detection difficult without advanced imaging and expert interpretation. Echocardiography is the primary diagnostic tool due to its safety, real-time imaging, and accessibility. However, the effectiveness of echocardiography heavily relies on the skill of the operator and interpreter, introducing variability and subjectivity into the diagnostic process.

In light of these limitations, there is growing interest in applying artificial intelligence—particularly deep learning models—to enhance the diagnostic process for CHD. These models aim to reduce reliance on specialist interpretation, improve diagnostic consistency, and make early detection more accessible, especially in resource-constrained environments. Deep learning approaches, including Knn,Gnb,Logistic Regression,Random Forare now being developed to analyze echocardiographic and cardiac MRI images for the identification and classification of CHD.

U-Net, a widely used architecture for medical image segmentation, is particularly effective in delineating cardiac structures such as ventricles, atria, and great vessels. Its encoder-decoder structure allows precise localization of abnormalities in complex pediatric heart anatomy. DenseNet and ResNet, known for their superior feature extraction and classification capabilities, have shown promise in identifying various CHD types by learning intricate patterns in cardiac imaging data. These models help automate the identification of atrial and ventricular septal

defects, Tetralogy of Fallot, coarctation of the aorta, and other critical anomalies.

However, the deployment of such systems is not without challenges. A significant limitation in CHD research is the scarcity of large, annotated pediatric datasets. The variability in heart morphology across different ages, stages of development, and types of CHD adds another layer of complexity that traditional machine learning techniques struggle to handle. Furthermore, rare CHD types may be underrepresented in available datasets, reducing the model's ability to generalize to unseen conditions.

To address these challenges, researchers are incorporating strategies such as transfer learning, which allows models pre-trained on large general image datasets to be fine-tuned on smaller, domain-specific datasets. Data augmentation is also employed to artificially increase the size and diversity of the training dataset, improving model robustness. Moreover, multi-modal fusion techniques that combine imaging data with clinical features—such as oxygen saturation, blood pressure, and genetic markers—have the potential to enhance predictive accuracy and provide more holistic diagnostic insights.

The real-world deployment of AI-driven CHD diagnostic systems holds the potential to revolutionize pediatric cardiology. In areas lacking sufficient medical infrastructure, these systems could assist general practitioners and community health workers in making initial assessments, thereby triaging high-risk patients for referral to specialized centers. Furthermore, the integration of cloud-based diagnostic platforms can facilitate remote consultation and telemedicine, expanding the reach of expert care.

Beyond diagnosis, these intelligent systems could support surgical planning by generating detailed anatomical maps and simulations, improving surgical outcomes and reducing operative time. Continuous learning mechanisms embedded in these models allow for ongoing improvement as new data is introduced, ensuring that diagnostic performance keeps pace with evolving medical standards and emerging CHD variants.

Ultimately, the success of such technologies will depend on interdisciplinary collaboration between clinicians, engineers, and data scientists, along with the creation of open-access datasets to encourage innovation and reproducibility. Regulatory approval, clinical validation, and ethical considerations—especially concerning pediatric data—must However, the deployment of such systems is not without challenges. A significant limitation in CHD research is the scarcity of large, annotated pediatric datasets. The variability in heart morphology across different ages, stages of development, and types of CHD adds another layer of complexity that traditional machine learning techniques struggle to handle. Furthermore, rare CHD types may be underrepresented in available datasets, reducing the model's ability to generalize to unseen conditions.

1.4 Objectives of the Project

This project aims to leverage advanced deep learning models to automatically analyze echocardiogram images, MRI, or CT scans to enhance the early detection and accurate diagnosis of Congenital Heart Disease (CHD) in pediatric and adult patients.

This project focuses on leveraging advanced deep learning techniques to automatically analyze medical imaging data—such as echocardiograms, MRIs, and CT scans—to enhance the early detection and accurate diagnosis of congenital heart disease in both pediatric and adult patients. Robust models like gnb,logistic regression,svm,decision tree,random tree will be employed for the precise classification and segmentation of structural cardiac abnormalities.

To improve diagnostic precision, the project incorporates sophisticated image preprocessing techniques including contrast enhancement, noise reduction, and motion artifact correction. These steps aim to improve imaging clarity and enable the extraction of clinically relevant anatomical features such as valve dimensions, chamber volumes, septal thickness, and patterns of blood flow.

Given the limited availability of labeled datasets, particularly in pediatric cases, the approach includes the use of transfer learning. Pre-trained models on large-scale or related medical imaging datasets (such as adult cardiac MRIs or even general datasets like ImageNet) will be fine-tuned to perform effectively on more specific and limited datasets.

Model performance will be rigorously evaluated using clinically relevant metrics such as sensitivity, specificity, area under the ROC curve (AUC), and the Dice coefficient for segmentation accuracy. The performance will also be compared across various imaging modalities and age demographics to ensure robustness.

1.5 Limitations of the Project

1. **Model Interpretability in Complex Structural Diagnoses**: CNNs and other deep learning models frequently operate as "black boxes," making it challenging to understand how certain

choices are made. In clinical settings, where explainability is essential for fostering trust and professional adoption, this lack of openness may provide difficulties.

2. **Diagnostic Errors: False Positives and False Negatives :** Although deep learning models trained on echocardiograms or MRIs may exhibit high accuracy, they are prone errors particularly in borderline or rare CHD cases. Mislabeling can be disastrous, either causing missed treatment windows (false negatives) or leading to unnecessary surgical consultations (false positives).

3.High Computational and Hardware Demands: Training and deploying deep learning models—especially those involving 3D imaging like cardiac MRIs or Doppler echocardiography—requires GPUs, high-memory servers, and real-time processing capabilities. This creates a barrier for low-resource hospitals or rural clinics, where CHD detection is often most needed.

4. Dependence on Imaging Quality and Consistency: Pediatric cardiac imaging is highly variable due to patient motion, operator skill, age-dependent anatomy, and different imaging modalities (2D echo, Doppler, fetal echo, MRI). Deep learning models can be highly sensitive to such inconsistencies, affecting performance.

5. Poor Generalization Across Populations and Devices: Poor Generalization Across Populations and Devices CHD incidence and presentation vary by genetic, geographic, and socioeconomic factors.Device-specific biases may cause inconsistent predictions across different hospitals or imaging vendors.

1.6 Organization of Documentation

The structure of the documentation is as follows:

Chapter 1: This chapter provides an in-depth introduction to the project, setting the stage for the entire documentation. It covers the key problem that the project aims to solve, highlighting the challenges faced in current communication systems for individuals with hearing impairments. The motivation behind the project is discussed, emphasizing the need for improved accessibility.Furthermore, the goals, scope, and importance of the project are outlined, providing a clear understanding of the project's significance in the context of global health and technology. This chapter also offers a brief overview of the project’s objectives and expected outcomes, giving readers a roadmap for the detailed discussions that follow.

Chapter 2: In this chapter, existing literature on communication accessibility for individuals with hearing impairments is reviewed. Key studies, research papers, and technologies that have been developed to address this issue are analyzed. This review helps identify the gaps in current solutions and highlights areas where further innovation is needed. The chapter also critiques various existing methodologies and technologies, comparing them to the proposed project approach. By assessing the strengths and weaknesses of prior work, this chapter sets the foundation for the project's methodology, illustrating how the proposed solution aims to overcome current limitations.

Chapter 3: Chapter 3 delves into the requirements and specifications of the project, outlining the precise needs that the solution must fulfill. The chapter includes a detailed analysis of the problem, breaking it down into smaller, manageable objectives. The methodology for gathering and analyzing these requirements is also discussed, showcasing the systematic approach used to ensure that the project remains focused on addressing real-world challenges. This chapter also explores the process of defining the project’s functional and non-functional requirements, which serve as the blueprint for the design and development phases. The importance of understanding the problem from a user-centered perspective is emphasized to ensure the project's success.

Chapter 4: This chapter focuses on the design factors of the project, elaborating on the various components and modules that make up the solution. It provides a detailed explanation of the system architecture, data models, and user interface design. The choices made in the design process, such as technology stack and platform selection, are discussed in detail. This chapter also explores the interaction between the different parts of the system and how they contribute to the overall functionality. The technical design choices are justified, ensuring that they align with the project's goals and requirements. By offering this level of detail, the chapter prepares readers to understand the technical complexities involved in the implementation phase.

Chapter 5: This chapter describes the programming languages, tools, and frameworks used to bring the project to life. It provides insights into why specific technologies were chosen based on their suitability for the project's needs. The chapter explains how the selected languages and tools were applied to build the system's components, from the back-end architecture to the user interface. It also highlights any unique challenges encountered during the development process and how they were overcome with the help of these tools. This chapter serves as a comprehensive guide to the technical stack, ensuring that readers understand the practical aspects of the project's implementation.

Chapter 6: This chapter focuses on the methods and strategies employed to test the system and verify its functionality. The testing process is broken down into several stages, including unit testing, integration testing, and user acceptance testing. The chapter elaborates on the techniques used to ensure that the system meets all its functional and non-functional requirements. It also discusses how user feedback was incorporated into the testing phase to improve usability. In addition to functionality, the chapter emphasizes the importance of testing for performance, security, and reliability to guarantee that the system can be deployed in real-world settings without issues.

Chapter 7: The final chapter of the documentation presents the overall conclusions and outcomes of the project. It reflects on the successes and challenges encountered during the project and discusses the main contributions of the system to improving communication accessibility for individuals with hearing impairments. The chapter summarizes the project's impact, providing a clear picture of how it

CHAPTER -2

LITERATURE REVIEW

2.1 Introduction

Congenital heart disease (CHD) remains one of the most prevalent and critical birth anomalies, contributing significantly to infant morbidity and mortality worldwide. Timely and accurate diagnosis is vital for effective intervention, yet conventional diagnostic techniques—primarily based on 2D echocardiography, fetal ultrasounds, and cardiac MRIs—require skilled interpretation by pediatric cardiologists. These traditional approaches are often limited by inter-observer variability, diagnostic delays, and unequal access to specialized care, especially in low-resource settings. To address these challenges, recent research has increasingly turned toward deep learning methods as a way to automate, standardize, and enhance the CHD diagnostic process.

The application of deep learning to CHD detection is primarily centered around analyzing echocardiographic and fetal ultrasound images. These modalities, while rich in diagnostic information, pose challenges such as image noise, variability in heart anatomy across age groups, and complex cardiac structures. Preprocessing techniques such as speckle noise reduction, contrast enhancement, normalization, and image augmentation have been shown to significantly improve the quality of input images and the performance of deep learning models. These steps are crucial for enabling the model to focus on relevant anatomical features and reducing the risk of overfitting, particularly when working with limited datasets.

Convolutional neural networks (CNNs) form the backbone of most deep learning-based CHD detection systems. Architectures like random forest,svm,gnb,knn have demonstrated high accuracy in identifying common heart defects such as atrial septal defects (ASD), ventricular septal defects (VSD), and more complex conditions like tetralogy of Fallot. These models excel at extracting spatial features from medical images, and when combined with recurrent layers or 3D CNNs, they can also capture temporal dynamics across echocardiographic video frames. Feature extraction remains a key challenge due to the subtle and diverse anatomical variations present in pediatric heart structures, especially during early gestational stages.

To overcome the scarcity of large annotated CHD datasets, many studies employ transfer learning, leveraging models pre-trained on general image databases like ImageNet. Fine-tuning these models on specific CHD datasets significantly boosts training efficiency and diagnostic accuracy.

2.2 Review of Congenital heart disease

Congenital heart disease (CHD) detection using deep learning has increasingly become a focus area in medical imaging research, driven by the growing need for accurate, scalable, and early diagnostic solutions. With CHD being the most common birth defect globally, affecting nearly 1% of live births, early detection is vital for timely intervention and improved survival outcomes. The use of deep learning—particularly Convolutional Neural Networks (CNNs)—has shown promise in addressing the limitations of conventional diagnostic methods such as fetal ultrasound and 2D echocardiography, which often require specialized expertise and are prone to inter-observer variability.

Several recent studies have demonstrated the potential of deep learning models in enhancing diagnostic precision for CHD. For instance, Zreik et al. used a CNN-based approach to classify congenital heart defects from fetal ultrasound images, achieving high sensitivity and specificity. Their model was trained using a combination of clinical echocardiographic datasets and public resources, showing the capability of deep networks to learn complex spatial patterns from cardiac structures. This work emphasizes the feasibility of deploying deep learning models to support early screening during prenatal care.

Arnaout et al. presented a landmark study using a deep learning pipeline trained on over 100,000 echocardiogram images to automatically detect congenital heart anomalies. Their system reached a diagnostic accuracy comparable to expert pediatric cardiologists, particularly in detecting anomalies such as hypoplastic left heart syndrome and tetralogy of Fallot. This study validated the real-world applicability of CNNs and highlighted the power of large datasets and robust annotation in training clinically reliable AI models.

In addition to standard CNN architectures, researchers have explored more advanced and efficient models like Knn,Gnb,Random forestfor CHD classification tasks. These models are valued for their ability to extract hierarchical features from noisy, low-contrast ultrasound or MRI images, a common challenge in pediatric cardiology. Transfer learning has been widely adopted in this domain due to the limited availability of large-scale CHD datasets. Pre-trained models on ImageNet or similar datasets are fine-tuned on smaller, specialized medical datasets, significantly improving convergence speed and model accuracy.

2.3 Review of Feature Extraction

Feature extraction is equally critical in the detection and classification of congenital heart disease (CHD), where identifying subtle, abnormal anatomical and functional patterns in fetal or pediatric cardiac imaging is essential. The complexity of CHD, which includes a diverse range of structural malformations such as septal defects, valve abnormalities, and chamber misformations, demands robust and precise feature extraction techniques to ensure accurate diagnosis. With the growing application of deep learning in cardiology, feature extraction has evolved from manual, handcrafted approaches to automated deep feature learning using advanced convolutional neural networks (CNNs).

In traditional CHD diagnosis, clinicians relied heavily on manual interpretation of echocardiographic images or cardiac MRI scans, often supported by geometric and Doppler-based measurements such as chamber size, flow velocity, and valve motion. While effective, these handcrafted features are time-intensive and require significant clinical expertise, which is not always available—especially in resource-limited settings. Deep learning-based models, however, automate this process by learning hierarchical feature representations directly from raw images, thus reducing reliance on domain-specific knowledge while maintaining diagnostic accuracy.

For instance, Arnaout et al**.** developed a CNN-based system that learned discriminative features from over 100,000 echocardiographic images to classify multiple CHD subtypes. The model autonomously identified patterns such as abnormal ventricular geometry and altered septal contours—features that are critical in diagnosing conditions like hypoplastic left heart syndrome and ventricular septal defect. These extracted features were not pre-specified but learned during training, showcasing the strength of deep learning in discovering clinically meaningful patterns that might be subtle or overlooked by the human eye.

2.4 Review of Encoding Techniques

Encoding techniques are equally essential in the field of congenital heart disease (CHD) diagnosis, particularly in the automated analysis of echocardiograms, fetal ultrasound images, and cardiac MRI scans. These techniques enable deep learning models to convert complex cardiac imaging data into structured representations that preserve the spatial, temporal, and morphological nuances of congenital abnormalities. Given the intricate structure of the heart and the subtle variations in CHD presentations, encoding plays a crucial role in enhancing the diagnostic accuracy and robustness of AI-driven tools.

Unlike lung cancer, where the focus is primarily on detecting and classifying nodules, CHD involves analyzing dynamic structures such as chamber size, valve motion, septal integrity, and blood flow patterns. To capture these features, encoding strategies must integrate both spatial encoding (e.g., anatomical structure) and temporal encoding (e.g., cardiac cycle motion). Arnaout et al. introduced a deep learning model trained on echocardiographic images for CHD classification, where feature encoding helped the model learn critical patterns such as ventricular asymmetry and malposition of the great arteries—key indicators of severe CHD forms like hypoplastic left heart syndrome (HLHS) and transposition of the great arteries (TGA).

One key technique used in CHD diagnostics is spatiotemporal encoding, often implemented through 3D-CNNs or recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) units. These models encode not just the anatomy but also motion dynamics, enabling the detection of abnormal flow patterns and arrhythmias. For example, Liang et al. developed a multi-frame CNN-LSTM model that encoded echocardiogram video sequences to diagnose septal defects and valve malformations with high sensitivity and specificity. The LSTM component encoded cardiac cycle phases, which helped in identifying delayed or abnormal valve closure—a hallmark of various congenital valve disorders.

Residual encoding strategies, are applied in CHD imaging to preserve important low- and mid-level features across deep layers. Zhang et al. applied a ResNet-based model to multi-view echocardiography, where encoding through skip connections retained edge details and chamber contours—features that are often lost in deeper convolutional layers.

Texture-based encoding is another valuable technique. In CHD detection, it helps to capture myocardial wall irregularities or valvular calcifications. Studies using Gray-Level Co-occurrence Matrices (GLCM) or Local Binary Patterns (LBP) have shown that encoding myocardial texture helps in classifying mild CHD cases, such as pulmonary stenosis, where the structural differences are not visually prominent. This encoded information is particularly beneficial in fetal ultrasound images,where resolution is lower, and subtle features are critical for diagnosis.

2.5 Objective and Scope of Present Work

Objective:

The primary aim of a deep learning–based system for congenital heart disease (CHD) diagnosis parallels that of lung cancer detection in its focus on enhancing diagnostic accuracy, efficiency, and accessibility, especially in settings where expert cardiologists or advanced imaging infrastructure may be scarce. This CHD project would center on leveraging advanced deep learning architectures, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to build a robust, intelligent system capable of automatically identifying and classifying various forms of CHD from echocardiographic or cardiac MRI images.

By automating the detection of structural and functional abnormalities—such as septal defects, valve malformations, and chamber enlargement—the system seeks to reduce diagnostic delays, support early interventions, and ultimately improve clinical outcomes for pediatric and adult patients alike. Given that CHD often manifests in early life and may require immediate medical or surgical intervention, timely and accurate diagnosis is critical for long-term survival and quality of life.

The project would incorporate cutting-edge image preprocessing techniques tailored for cardiac imaging, such as frame selection, motion compensation, and contrast enhancement. These methods enhance the clarity of critical anatomical structures, allowing for more reliable feature extraction. Transfer learning from pre-trained networks (e.g., ResNet-50 or EfficientNet-B0) would be applied and fine-tuned on domain-specific datasets, enabling the model to efficiently adapt to cardiac image characteristics and reduce training time, especially when data is limited.

A major innovation would be the integration of temporal encoding to capture dynamic heart motion across cardiac cycles—vital for detecting rhythm-based abnormalities and valve function irregularities. This could be achieved using LSTM networks or 3D-CNNs to process echocardiogram video sequences, facilitating real-time CHD detection in clinical and screening environments.

Finally, by integrating explainable AI (XAI) tools like Grad-CAM, clinicians could visualize how the model reaches its conclusions—building trust in AI-assisted decisions and supporting clinical validation. The ultimate goal is to reduce misdiagnosis, enable early surgical planning or medical therapy, and support lifesaving interventions for congenital cardiac anomalies.

Scope:

Image Preprocessing and Enhancement: Preprocessing is a critical step in preparing cardiac imaging data for deep learning analysis. In the case of congenital heart disease, imaging modalities such as echocardiograms and cardiac MRIs often contain artifacts like motion blur, speckle noise, and inconsistent contrast due to patient movement or machine variability.

Integration of Deep Learning Models: The core of the project involves the application of powerful convolutional neural network (CNN) architectures Gnb,Knn,Logistic Regression,Decision Tree to detect and classify various forms of congenital heart disease.

Automated Feature Extraction and Classification: One of the key objectives of this system is to automate the extraction of clinically meaningful features from heart images. Using deep learning, the model identifies critical anatomical and functional parameters, including chamber size, wall motion, valve morphology, and blood flow dynamics (if Doppler imaging is available).

Training, Testing, and Validation: For effective detection of congenital heart disease (CHD) using deep learning, systematic training, testing, and validation processes are critical to ensure model reliability and clinical applicability.

Model Optimization and Generalization: In congenital heart disease (CHD) detection, enhancing model generalization is crucial due to the wide anatomical variability across patient age groups and types of cardiac anomalies.

User Interface and Result Visualization: An effective AI-based CHD diagnosis tool must be interpretable and user-friendly, especially for cardiologists, pediatricians, and sonographers. The user interface is designed to present predictions with annotated cardiac images, highlighting suspected malformations or septal defects using visual overlays.

Future Enhancements: Effective CHD model development depends heavily on clean, well-annotated cardiac imaging datasets. The project will establish systematic data management protocols, including consistent annotation of CHD types, standardization of metadata (age, heart rate, modality), and robust version control of imaging sets.

Data Management and Annotation: Implementing structured data management practices, including consistent labeling, metadata organization, and version control for datasets to ensure reproducibility and facilitate future expansions.

Clinical Collaboration and Validation: Engaging with radiologists and oncologists for feedback on model predictions and system usability. Conducting pilot studies in a clinical setting to validate practical applicability.

Documentation and Deployment Readiness: Creating thorough documentation covering model architecture, preprocessing pipelines, and usage guidelines. Preparing the solution for deployment in clinical environments with proper testing and modular design.

2.6 Existing System

The existing systems for congenital heart disease (CHD) detection largely depend on conventional clinical assessments and echocardiographic evaluations, with relatively limited integration of advanced automated deep learning-based solutions.

1. **Manual Interpretation Required** :Cardiologists and sonographers must visually examine complex heart structures in ultrasound or echocardiography, requiring high expertise

studies using conventional models.

2. **Time-Consuming Process** : Manual image analysis increases diagnostic time, especially during busy clinical workflows.

3. **High Dependence on Operator Skill** – Diagnostic accuracy is directly tied to the clinician's experience and training.The standard medical imaging format (DICOM) presents difficulties in processing, which contributes to inefficiency and increased implementation costs in clinical setups.

4. **Human Error Risk** – Subtle structural anomalies, especially in early fetal development, are often missed

5. **Inconsistent Diagnostic Accuracy** – Results vary across institutions and between operators due to equipment and skill variability.

6. **Limited Prenatal Screening** – Many heart defects are not detected until after birth, reducing early intervention opportunities.

7. **Fetal Imaging Challenges** – Movement, suboptimal positioning, and limited imaging windows complicate prenatal echocardiography

8. **Delayed Postnatal Diagnosis** – Lack of universal newborn screening delays identification of critical CHD cases.

9. **Shortage of Pediatric Cardiologists** – Many regions, especially in LMICs, face a critical shortage of trained specialists.

10. **High Equipment Costs** – Advanced imaging machines are expensive and inaccessible in many healthcare settings.

11**.Uneven Geographic Access** – Rural and underserved areas lack the infrastructure for routine screening.

12**.Non-Standardized Protocols** – There is no global standard for CHD screening during pregnancy or infancy.

13**.Limited Use of AI Tools** – Although promising, AI-based diagnostic tools have yet to be widely adopted in clinical practice.

14**.Lack of Real-Time AI Support** – Most AI tools do not yet assist clinicians during live echocardiographic scanning.

15**.Data Silos** – Imaging data, clinical records, and genetic profiles are fragmented across incompatible systems.

16.**Scarcity of Annotated Datasets** – Developing effective AI models for CHD detection requires large volumes of high-quality, labeled echocardiographic or imaging data. However, such datasets are rare due to patient privacy concerns, variability in imaging protocols, and the labor-intensive process of expert annotation, limiting the performance and generalizability of deep learning algorithms.

17.**Poor System Integration** – Most healthcare institutions use fragmented software systems that are not designed to integrate AI tools seamlessly into clinical workflows. This lack of interoperability between imaging platforms, electronic health records, and AI algorithms hinders the potential for real-time, data-driven support during patient evaluations.

18.**Low AI Explainability** – Many AI models operate as "black boxes," offering predictions without clear reasoning or visual justification. In critical pediatric cases like CHD, clinicians are hesitant to act on AI outputs that they cannot verify or interpret, making trust and adoption a major challenge in real-world settings.

19. **Regulatory Barriers** – The implementation of AI in clinical environments faces strict regulatory oversight to ensure safety and efficacy. Lengthy approval processes, inconsistent global standards, and concerns over liability and accountability delay the integration of AI-based CHD diagnostic tools into everyday practice.

20**.Clinical Resistance to Change** – Despite technological advances, many healthcare professionals remain cautious about adopting AI systems due to concerns over job displacement, workflow disruption, and loss of clinical autonomy. This resistance can slow innovation and limit the impact of advanced solutions on CHD care delivery.

2.7 Proposed System

By applying deep learning algorithms to thoracic CT scan pictures, the proposed system presents a sophisticated, AI-driven method for enhancing the early identification and categorization of lung cancer. It seeks to provide a dependable, automated, and scalable diagnostic tool in order to overcome the current drawbacks of human radiological diagnosis, including inter-observer variability and delays. The key components of the proposed system include:

1. Deep learning algorithms can automatically analyze echocardiograms to detect congenital abnormalities, reducing dependence on expert interpretation. These models identify patterns in heart structure and function that may be missed in manual evaluations. Early diagnosis is crucial for timely intervention in infants and children.

2. 3D CNNs applied to cardiac MRI and CT scans enable precise localization of structural defects like septal anomalies and valve malformations. These models process volumetric data to understand spatial relationships in the heart. This aids in accurate surgical planning and diagnosis.

3. Transfer learning using pre-trained models like EfficientNet and DenseNet helps overcome the challenge of limited pediatric cardiac datasets. Fine-tuning these models boosts their performance on CHD-specific tasks. This approach saves training time while maintaining high diagnostic accuracy.

4. Preprocessing steps such as image normalization, noise reduction, and resizing are essential for preparing cardiac images. These steps ensure consistent quality and improve the model's ability to detect subtle heart defects. Preprocessing also enhances the efficiency of feature extraction

5. Integration with hospital imaging databases and PACS (Picture Archiving and Communication Systems) for real-time analysis and reporting, enabling seamless collaboration between radiologists and AI-based decision support tools.

6. The system will standardize CT scan protocols across different healthcare facilities to ensure consistency in image quality, reducing discrepancies that may arise from variations in imaging techniques and equipment.

7. The proposed system will integrate with cloud-based platforms for secure, scalable storage and

easy access to medical images, enabling healthcare professionals to retrieve and share diagnostic data efficiently from remote locations.

8. The system will include a real-time monitoring feature that automatically alerts radiologists and healthcare providers of potential lung cancer indicators, ensuring timely intervention and reducing the likelihood of misdiagnosis or missed cases.

9. The system will allow integration with patient history records, including previous scans, medical reports, and risk factors, to provide a more comprehensive understanding of the patient's condition and facilitate better-informed clinical decisions.

10. A streamlined and intuitive user interface will be developed for radiologists to interact with the system, enabling quick and easy navigation through patient images, analysis results, and diagnostic reports while minimizing the learning curve.

CHAPTER-3

ANALYSIS

3.1 Introduction

This project centers on advancing the early detection and classification of Congenital Heart Disease (CHD) through the application of deep learning technologies, with the aim of enhancing diagnostic accuracy, efficiency, and clinical accessibility. By leveraging high-resolution cardiac imaging modalities such as echocardiography, cardiac MRI, and CT, the system incorporates cutting-edgeKnn,Gnb,Logistic Regression,Decision Tree,Random forest,svm—to automatically detect and characterize congenital anomalies in cardiac structures. The platform employs advanced preprocessing techniques like spatial normalization, noise reduction, and contrast enhancement to standardize input data and facilitate more accurate feature extraction and classification.

Despite the challenges posed by variability in pediatric cardiac anatomy and limited annotated datasets, the project has demonstrated significant progress in reliably identifying key CHD conditions, such as septal defects, transposition of great arteries, and tetralogy of Fallot. Automated segmentation and classification have yielded promising results in distinguishing between different CHD subtypes, thereby supporting its potential for integration into routine clinical practice. Key performance metrics, such as precision, sensitivity, and F1 scores, indicate the system’s utility in both initial screenings and secondary evaluations by specialists

In addition to improving diagnostic accuracy, the system is purposefully designed to reduce the burden on pediatric cardiologists by automating time-intensive aspects of diagnosis. This allows clinicians to prioritize high-risk cases and allocate resources more effectively, ultimately accelerating the diagnostic process and improving patient outcomes. Cross-validation with diverse datasets—spanning different imaging devices, institutions, and demographics—has been performed to ensure generalizability and robustness of the model across real-world scenarios.

Looking forward, future enhancements aim to integrate real-time diagnostic capabilities with bedside echocardiography systems and mobile health platforms. This would enable early CHD screening even in under-resourced or remote healthcare settings, addressing gaps in pediatric cardiac care delivery

3.2 Requirement Specification

3.2.1 Hardware Requirement

This project is intentionally engineered to operate efficiently across a broad spectrum of commonly available hardware platforms, making the CHD diagnostic system accessible to users regardless of their technological infrastructure. The design philosophy prioritizes inclusivity and usability, ensuring that medical professionals, caregivers, and institutions with limited resources can fully benefit from the system’s capabilities without needing high-end equipment.

The system is optimized to run smoothly on standard desktop computers and laptops that meet modest contemporary computing specifications. It also extends support to mobile platformssuch as smartphones and tablets—enabling pediatric cardiologists and clinicians to conduct on-the-go assessments, particularly in remote or under-resourced environments. By focusing on cross-device compatibility, the system ensures real-time usability in varied settings, including mobile clinics, rural hospitals, and educational institutions.

For optimal functionality, the recommended hardware includes a modern multi-core processor like Intel Core i5 or AMD Ryzen 5, accompanied by at least 8GB of RAM. These specifications are sufficient for tasks such as loading cardiac images, running pre-trained models, and interacting with the interface in real time. The storage requirements remain minimal, typically limited to installation files, temporary scan data, and optional logs or reports generated during diagnostics. The system’s lightweight design guarantees responsiveness without overwhelming the hardware, ensuring smooth performance even on older machines.

To enhance functionality in advanced use cases—such as CHD subtype classification using full 3D cardiac MRI volumes or real-time echocardiographic video processing—more powerful setups are recommended. These include processors like Intel Core i7 or AMD Ryzen 7, with 16GB or more of RAM and dedicated GPUs (e.g., NVIDIA GTX or RTX series) with at least 4GB of VRAM. GPU acceleration is particularly beneficial for training custom models or processing large volumes of pediatric cardiac imaging data, offering substantial improvements in computation speed and efficiency

.

Optional hardware components can further enrich the user experience, especially in integrated environments. For example, portable ultrasound probes or hand-held echocardiography devices can connect to mobile devices running the system, enabling point-of-care diagnostics. However, these peripherals are not essential for basic diagnostic functionality and are considered enhancements rather than requirements, allowing the system to remain accessible to users with basic setups.

Finally, the system's ability to run on a wide variety of devices without compromising performance is essential for its adoption in diverse environments. Whether the user is working on a personal laptop, using a tablet in the field, or accessing the system on a shared public computer, it ensures inclusivity and ease of access. Furthermore, by minimizing hardware dependency, the system lowers the barrier to entry for individuals with varying access to high-end devices, making it suitable for educational institutions, small businesses, and non-profit organizations that may not have access to the latest hardware but still wish to leverage the benefits of the technology.

3.2.2 Software Requirement

The software requirements for this project are centered around tools and frameworks that support efficient development, implementation, and deployment of deep learning models for lung cancer detection. Python is the primary programming language used, offering robust support for machine learning and image processing. Development and experimentation are carried out using Jupyter Notebook, which facilitates interactive coding and model testing. Streamlit is employed to build a user-friendly web interface for deploying the trained models, enabling real-time predictions and visualizations. Visual Studio is used as an integrated development environment (IDE) for managing and organizing the project’s codebase. These tools collectively ensure seamless integration of preprocessing, model training, and deployment components, enabling the creation of a reliable and accessible diagnostic application.

The project also utilizes a variety of deep learning frameworks to support model training and evaluation. TensorFlow and Keras are the primary libraries used for building and training the neural networks.

TensorFlow provides flexibility and scalability for handling complex models, while Keras simplifies the process of designing and training deep learning models with its high- level interface. For image processing and augmentation, OpenCV and Pillow are integrated into the workflow, allowing for efficient preprocessing and manipulation of medical images. The combination of these frameworks ensures that the models can be trained with optimal performance while maintaining a focus on accuracy and reliability in detecting lung cancer.

In addition to the development tools, version control and collaboration are crucial for the project's success. Git is used for tracking changes to the codebase and ensuring that the development process remains organized and transparent. GitHub serves as the platform for code hosting and collaboration, enabling multiple developers to work on the project concurrently and manage contributions efficiently. Furthermore, Docker is employed to containerize the application, ensuring that it can be deployed in any environment without dependency issues.

To ensure seamless data management and accessibility, the project leverages cloud storage services, specifically for storing and retrieving medical image datasets. Cloud storage solutions like Amazon S3 or Google Cloud Storage are integrated into the workflow, providing a secure and scalable method for managing large datasets typically used in medical imaging. These platforms facilitate easy access to the data, ensuring that it is always available for model training, evaluation, and testing. Additionally, the integration with cloud storage enhances the project’s ability to scale by enabling quick sharing and retrieval of resources across different computing environments, which is crucial in collaborative settings or when working with large volumes of medical imaging data.

The project also incorporates testing and validation procedures to ensure that the models are robust and reliable. Unit testing and integration testing are carried out using frameworks like pytest and unittest to verify the functionality of individual components of the system. These testing frameworks help in identifying potential issues during the development process, enabling developers to address them before deploying the system. Furthermore, cross-validation techniques are applied during model training to assess the model's performance on unseen data, ensuring generalization and reducing the risk of overfitting.

For model deployment, the project utilizes continuous integration and continuous deployment (CI/CD) pipelines to streamline updates and ensure consistent delivery of new features or bug fixes. Jenkins or GitHub Actions can be configured to automate the deployment process, from code updates to model retraining. This automation minimizes manual errors, accelerates the release cycle, and ensures that the deployed model always reflects the latest improvements. Additionally, for monitoring model performance in production, tools like Prometheus and Grafana are used to track system health, performance metrics, and potential issues, allowing for proactive maintenance and quick intervention when necessary. This complete deployment pipeline ensures that the application remains reliable, efficient, and ready for clinical use.

CHAPTER-4

DESIGN

4.1 Introduction

The software infrastructure for this Congenital Heart Disease (CHD) diagnostic system is carefully selected to support efficient development, deployment, and continuous improvement of deep learning models tailored to pediatric and adult cardiac imaging. The core programming language is Python, chosen for its rich ecosystem in machine learning, signal processing, and medical image analysis. Development and experimentation are primarily carried out using Jupyter Notebook, enabling interactive exploration, model testing, and visualization of complex cardiac structures. Streamlit is used to build a streamlined, intuitive web interface that allows clinicians to upload cardiac images and receive real-time diagnostic predictions and visual interpretations. Visual Studio Code (VS Code) is employed as the integrated development environment (IDE) to manage the growing codebase, enhance productivity, and support debugging and version control integration

To build, train, and validate the deep learning models, the system leverages TensorFlow and Keras, which offer scalable architecture and user-friendly APIs for rapid model prototyping and training. These frameworks are particularly suited for implementing Convolutional Neural Networks (CNNs) and recurrent models used in dynamic image sequences like echocardiograms. Image preprocessing and augmentation are handled using OpenCV and Pillow, facilitating tasks such as resizing, noise reduction, and spatial alignment of cardiac imaging modalities (e.g., 2D/3D echo, cardiac MRI). Together, these tools ensure efficient handling of heterogeneous medical data while maintaining diagnostic accuracy and reproducibility.

Version control and collaborative development are integral to managing this multi-phase project. Git is used to track changes, maintain development branches, and support modular updates, while GitHub provides a centralized platform for collaboration, issue tracking, and deployment history. Containerization using Docker ensures that the entire software stack—ranging from libraries to runtime environments—can be deployed consistently across machines, reducing compatibility issues and simplifying rollout in clinical or research settings.

Cloud-based infrastructure plays a vital role in enabling access to large-scale, annotated cardiac imaging datasets. Platforms such as Google Cloud Storage or Amazon S3 are integrated to store, manage, and retrieve echocardiography, MRI, or CT images for model training and inference. These services ensure secure data access across distributed teams and facilitate real-time processing in remote environments, such as rural clinics or mobile cardiac units. The cloud also allows seamless scaling of compute resources for training deep learning models with large, high-resolution datasets.

Robust testing and validation mechanisms are built into the development pipeline to ensure model safety and clinical relevance. Unit and integration testing frameworks like pytest and unittest are employed to validate software modules—ranging from image loaders to prediction interfaces. During model development, k-fold cross-validation and stratified sampling techniques are used to assess generalizability across diverse patient populations and imaging conditions, addressing key challenges such as data imbalance and patient variability in congenital heart defects.

For deployment, the system is integrated into a CI/CD pipeline using tools like GitHub Actions or Jenkins, automating stages such as model retraining, application updates, and interface enhancements. This ensures timely delivery of improvements without disrupting ongoing clinical use. Performance monitoring tools like Prometheus and Grafana are incorporated to track inference latency, usage patterns, and system health in real-time, allowing proactive intervention in case of anomalies or system degradation

4.1 System Architecture

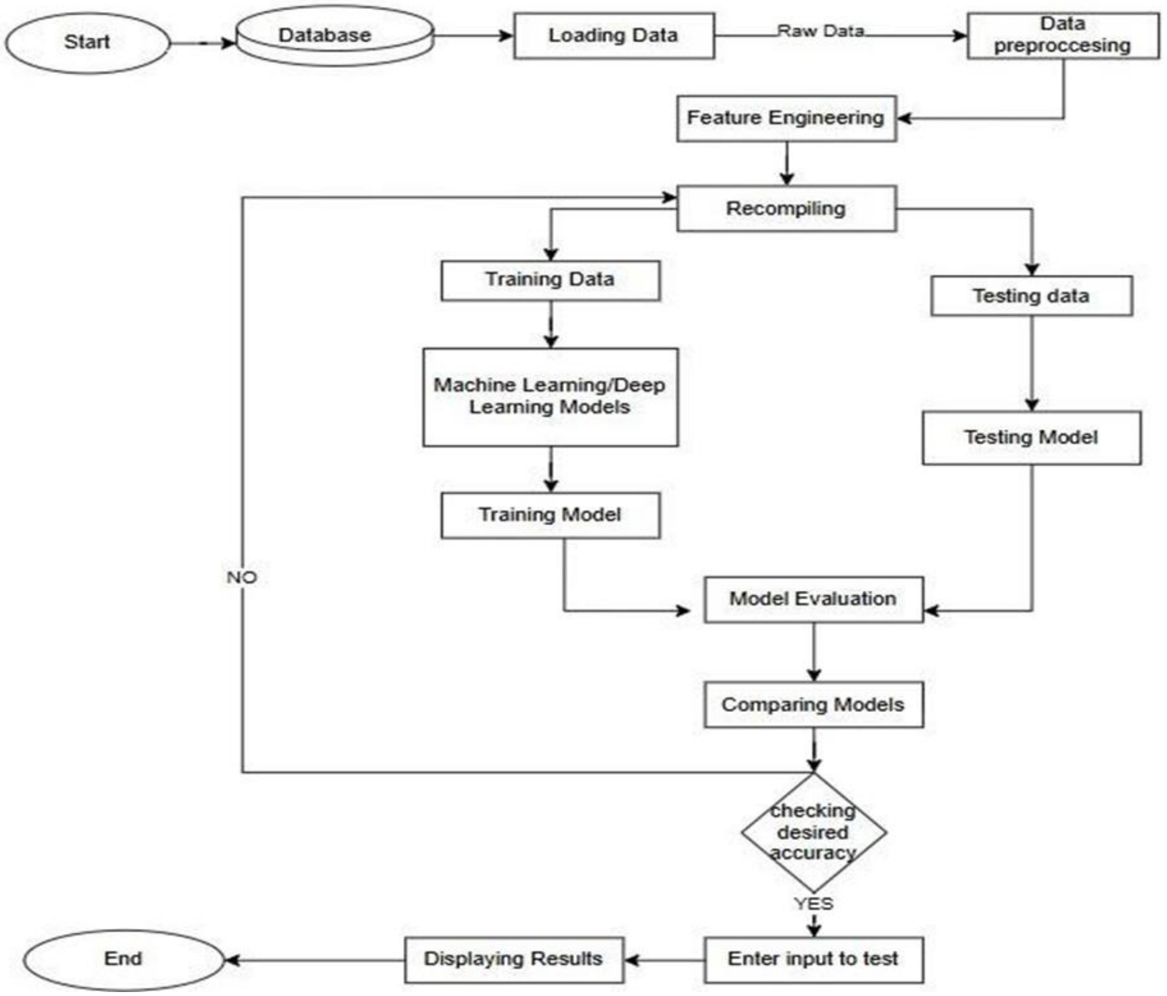


Figure 4.1: System Architecture of the Application

The Congenital Heart Disease (CHD) detection system is built on a modular, end-to-end architecture tailored to automate and streamline cardiac diagnosis. The design integrates multiple components—data acquisition, image preprocessing, model training, and diagnostic prediction—into a cohesive pipeline. This structure ensures secure data handling, efficient processing, and real-time diagnostic support, ultimately enhancing clinical decision-making. Its modularity also supports system scalability, adaptation to evolving diagnostic standards, and integration with existing hospital information systems

The pipeline begins with the data acquisition phase, focusing on collecting echocardiograms, cardiac MRIs, or CT angiograms, depending on the type of congenital defect being investigated. These images are sourced from hospital PACS or open-access cardiac databases and are then anonymized to protect patient identity. Since CHD encompasses a wide range of structural anomalies, capturing high-resolution and multi-view datasets is essential for model accuracy. Data diversity is emphasized to ensure the system is applicable across age groups, from neonates to adults.

Once acquired, images undergo preprocessing, including normalization of pixel intensity, contrast enhancement, resizing, and artifact removal. This is particularly critical in pediatric cardiology, where anatomical structures are small, dynamic, and highly variable. Additionally, temporal sequence alignment is performed for video-based echocardiographic data to synchronize cardiac cycles. These refined images provide the consistency necessary for robust feature extraction and classification.

The preprocessed data is systematically split into training, validation, and test sets to support the development of a generalizable model. During the model training phase, advanced CNN-based architectures such as ResNet-50, VGG19, and EfficientNet-B0 are deployed. These models are fine-tuned using transfer learning techniques on annotated datasets curated by cardiologists. ResNet-50, known for its residual learning, proves especially effective in capturing intricate structural variations like septal defects, coarctation, or valve malformations.

After training, the model proceeds to the prediction phase, where it evaluates unseen cardiac images or video loops and classifies them based on the presence and type of congenital anomaly. Results are presented through a user interface, offering visual overlays (e.g., bounding boxes or heatmaps) and confidence scores to assist cardiologists in interpreting findings. This stage minimizes diagnostic delays and reduces the likelihood of oversight, particularly in resource-constrained or high-volume clinical environments

User Interaction

Users interact with the system via a user-friendly interface that allows for the submission of CT scan images or related health data. These users can be healthcare professionals seeking a second opinion or patients monitoring their condition. The interaction triggers the backend processes that begin with data processing. The system is designed to provide a seamless and intuitive experience, minimizing complexity for users. It ensures fast input handling and prompt feedback, supporting clinical efficiency.

The system also offers various tools for healthcare professionals to review and analyze results in detail. These include options for adjusting image contrast, zooming into specific areas of interest, and viewing side-by-side comparisons of different scan slices. The interface is designed to be intuitive, providing easy navigation and allowing users to quickly access relevant patient history or prior scans. For patients, the system can display simplified, easy-to-understand reports, ensuring that they are informed about their health status without being overwhelmed by complex medical terminology.

Additionally, the system is designed to support collaboration among multiple healthcare providers. It enables real-time sharing of patient data and scan results, allowing specialists from different fields to collaborate on diagnosis and treatment planning. This collaborative feature promotes a more accurate and holistic approach to patient care. To further enhance user experience, the system is optimized for performance, ensuring that it operates smoothly across devices and can handle large datasets without lag or delays, contributing to a more efficient and effective healthcare workflow.

The system incorporates robust user authentication and data privacy mechanisms to ensure that sensitive medical information remains secure. Only authorized personnel can access patient records, and every access attempt is logged for accountability. The platform adheres to healthcare regulations like HIPAA, ensuring that data sharing and storage comply with legal and ethical standards. Users can set permission levels, enabling doctors, radiologists, and support staff to access only the data relevant to their roles. Patients can also control who sees their information, fostering trust and transparency in the system.

Moreover, the user interface supports customization based on individual preferences and roles. Healthcare professionals can personalize dashboards to highlight the most relevant patient data, preferred tools, and scan types. The system provides alerts and reminders for scheduled scans, follow-up appointments, and report generation, helping users stay on top of their tasks. For patients, the system can send notifications regarding upcoming procedures or new results, enhancing their engagement in the care process. This level of personalization and automation contributes significantly to streamlining operations and improving the overall user journey.

Data Source and Storage

The data utilized in the CHD detection system is sourced from reputable medical repositories and partner cardiology institutions. These datasets typically consist of annotated echocardiograms, cardiac MRIs, and CT angiograms, which are critical for training, validating, and testing deep learning models. Each case includes expert-labeled structural anomalies—such as septal defects, valve malformations, or outflow tract obstructions—validated by pediatric and adult cardiologists. Structured storage ensures systematic access, while compliance with data protection regulations like HIPAA or GDPR is strictly maintained.

The primary datasets are drawn from publicly available and institutional sources such as the CAMUS dataset (echocardiography) or the UK Biobank cardiac MRI repository. These datasets are carefully curated to ensure that they represent various congenital heart conditions across age groups, from neonates to adults. Annotations are done manually by trained cardiologists, marking regions of interest and providing labels for defects such as ventricular septal defects, atrial septal defects, Tetralogy of Fallot, and others. The inclusion of metadata—like patient age, symptoms, and diagnostic outcome—further enriches the dataset for clinical relevance.

Given the volume and resolution of cardiac imaging data, a cloud-based storage solution is employed to manage terabyte-scale datasets efficiently. The infrastructure supports rapid access and parallel processing during training and inference. Data is encrypted both at rest and in transit, with fine-grained access controls and authentication mechanisms in place. Frequent backups and real-time replication ensure availability and resilience, while audit trails track access and usage to support traceability and security.

In addition to the primary dataset, supplementary data sources are utilized to enhance the diversity and robustness of the training process. These supplementary datasets may include publicly available collections of CT scans or data from specific medical institutions. This helps broaden the scope of the models, ensuring they can generalize across various lung conditions and patient demographics. The integration of multiple datasets also mitigates any potential biases that may arise from using a single dataset, making the models more reliable for a broader range of real-world scenarios.

Data preprocessing plays a crucial role in the success of the deep learning models. Once the CT scan images are collected, they undergo rigorous preprocessing steps to ensure that they are suitable for training. This includes image normalization, resizing, and augmentation techniques such as rotation, flipping, and zooming. These preprocessing steps help improve the model's robustness by increasing the variability of the images, preventing overfitting, and enhancing the model's ability to identify features that are crucial for accurate diagnosis. The use of high-quality images combined with these preprocessing techniques ensures that the deep learning models are trained on the best possible input data.

To maintain the highest standards of data management, regular audits and monitoring of the database are conducted. These audits ensure that all data storage practices comply with relevant healthcare data protection laws and regulations. Any updates to the dataset are carefully tracked, and version control mechanisms are in place to ensure that the models are trained on the most current and relevant data. Additionally, the database is monitored for any unauthorized access attempts, and access logs are maintained to trace any discrepancies. By adhering to these strict protocols, the system ensures the privacy and security of patient data, while also supporting the continued improvement and refinement of the models.

Data Preprocessing

Preprocessing in congenital heart disease (CHD) diagnosis is a vital step that ensures cardiac imaging data, such as echocardiograms, MRIs, and CT angiograms, are clean, standardized, and suitable for deep learning-based analysis. This process involves denoising, temporal alignment, spatial normalization, and segmentation of relevant cardiac structures like ventricles, atria, valves, and vessels. Effective preprocessing plays a major role in the model’s ability to learn and identify structural abnormalities that define CHD.

Raw cardiac images often suffer from noise, motion artifacts, and probe-induced distortions, especially in echocardiograms. To address these challenges, filters such as Gaussian and median are applied to smooth the images, while speckle-reducing diffusion techniques help maintain important edges. For time-series data like cine-MRI, temporal smoothing techniques are applied to stabilize cardiac motion across frames, allowing better interpretation of heart dynamics.

To standardize inputs for training, cardiac images are resized to fixed dimensions, ensuring consistent input size for the deep learning models. Pixel values are normalized to a common range to help the model converge during training and focus on relevant anatomical differences. Additionally, frames in cine or Doppler sequences are aligned to specific phases of the cardiac cycle, such as end-diastole or end-systole, based on ECG signals or time-based markers to capture consistent anatomical view

One of the most important preprocessing steps in CHD diagnosis is segmentation, which isolates key cardiac regions such as the left ventricle, right atrium, aorta, or septum. Segmentation helps the model concentrate on structural patterns that are clinically significant. Deep learning-based segmentation models, such as U-Net and its variants, are commonly used to extract precise cardiac structures, while simpler methods like thresholding or region growing may be employed in more straightforward 2D echo images. This ensures that only relevant regions are used for training and prediction.

Another essential aspect of preprocessing is the augmentation of training data. Medical datasets often suffer from a lack of diversity due to the limited number of available CT scans. To overcome this, data augmentation methods such as rotation, flipping, zooming, and shifting are applied to artificially expand the dataset. This approach increases the model's ability to generalize and prevents overfitting, as the model is trained on a more diverse set of images. Augmentation is particularly useful when dealing with imbalanced datasets, where certain features (like small nodules) might be underrepresented, as it ensures that the model encounters a broader range of examples during training.

Expert Review (Doctor/Members)

Medical professionals play a key role in validating the predictions made by the system. Their expertise helps ensure that the automated results align with clinical understanding. This collaborative model improves trust in the system and allows for real-world feedback to fine-tune performance. Doctors may also assist in labeling datasets, contributing to better model training. Their involvement is critical to bridge the gap between AI predictions and human diagnosis.

Medical professionals are essential in reviewing the system's predictions, as they provide the clinical insight needed to confirm or challenge the results. Their deep understanding of medical conditions, patient history, and current treatment protocols helps ensure that automated suggestions align with real-world clinical practices. This expert validation helps reduce errors, ensuring that the system's outputs are not only accurate but also actionable in a healthcare setting. By cross-referencing automated predictions with their knowledge, doctors can identify potential issues, enhancing patient safety.

Moreover, doctors' input extends beyond validation to improving the system's performance. Their involvement in labeling medical data, reviewing cases, and providing clinical context allows the system to evolve and better understand complex medical scenarios. They also play a key role in the continuous education of the system by offering feedback on its predictions and suggesting improvements. Their expertise ensures that the system remains relevant, up-to-date, and effective in the ever-evolving medical field.

Medical professionals are integral to the process of refining diagnostic systems by providing real-world expertise. Their knowledge of patient symptoms, medical history, and evolving conditions allows them to assess whether the system's recommendations are consistent with clinical practices.

Doctors' deep understanding of the intricacies of human health makes them key figures in detecting discrepancies or errors that may arise in the automated analysis, ensuring that the recommendations align with accepted medical standards and protocols. This collaboration ensures that technology is not only accurate but also relevant in the clinical environment.

In addition to validation, doctors contribute significantly to the development of clinical decision support systems through their participation in ongoing research and development. By analyzing patient data and recognizing patterns that the system may not fully comprehend, they offer valuable insights into disease progression, treatment responses, and patient outcomes. Their clinical expertise helps steer the system's focus toward the most relevant aspects of medical care, ensuring that the technology adapts to the nuances of patient care. As medical research continues to evolve, doctors play a pivotal role in maintaining the system's ability to keep pace with new discoveries and treatment methods.

Doctors also foster collaboration between medical technology developers and healthcare providers, ensuring that the system's implementation is practical and effective in clinical settings. Their input helps developers understand the unique challenges faced by healthcare professionals, such as time constraints, resource limitations, and the variability of patient conditions. By providing direct feedback on usability, doctors help ensure that the system is designed to be intuitive, user-friendly, and adaptable to different clinical environments. Their involvement ensures that the system is not only scientifically sound but also practically applicable in real-world medical practice, improving both patient outcomes and healthcare efficiency.

Data Splitting

Data splitting in congenital heart disease (CHD) diagnosis is a crucial step in ensuring that deep learning models trained on medical imaging and clinical data are robust, generalizable, and clinically reliable. Given the inherent complexity and variability in CHD conditions—ranging from simple septal defects to complex multi-chamber malformations—careful partitioning of data into training, validation, and test sets becomes essential for building accurate diagnostic tools. This structured approach helps prevent the model from overfitting to particular anatomical presentations or imaging conditions and ensures it can effectively analyze diverse patient cases.

A well-executed data split also helps in mitigating the risk of overfitting, where the model performs exceptionally well on the training data but poorly on new data. Overfitting occurs when the model becomes too complex, capturing noise or irrelevant details rather than generalizable patterns. By validating the model on a separate dataset and testing it on a distinct test set, the chances of overfitting are significantly reduced, ensuring that the model's performance reflects its ability to generalize to real-world scenarios. Proper data splitting, therefore, is essential for the robustness and reliability of any machine learning application.

In the context of CHD, datasets often consist of echocardiograms, cardiac MRIs, CT scans, and sometimes even time-series physiological signals. These datasets may be limited in size due to the rarity of certain CHD subtypes and the ethical considerations around pediatric patient data collection. Therefore, partitioning the data correctly becomes even more critical. Typically, 70% of the dataset is allocated for training, where the model learns to identify patterns in cardiac anatomy and function. Around 15% is used for validation, which serves to fine-tune model parameters and monitor learning progress. The remaining 15% is reserved for testing, where final model performance is evaluated on unseen data to simulate real-world clinical application

Due to the rarity and heterogeneity of CHD cases, stratified sampling during data splitting is often necessary to ensure that all significant CHD types are proportionally represented in each subset. This prevents the model from becoming biased toward more common conditions, such as atrial septal defects, while underperforming on less frequent but clinically critical anomalies like transposition of the great arteries. Maintaining this balance across data partitions allows the model to develop a more comprehensive understanding of the various manifestations of CHD

The validation set plays an essential role in guiding model development by offering real-time performance feedback during training. For CHD, where precise localization of structural anomalies is vital, the validation set helps determine if the model is correctly identifying clinically relevant features, such as chamber enlargement, valve regurgitation, or abnormal blood flow paths. If the model shows a performance gap between the training and validation sets, it may be overfitting or underfitting. Such feedback can inform corrective actions like adjusting the model architecture, applying regularization, or augmenting the trainin

Deep Learning Model Prediction

Advanced deep learning models such as CNNs are employed to analyze processed CT scans and identify potential cancerous patterns. These models have been trained to detect features specific to malignant and benign nodules. Based on input data, the system predicts the likelihood of lung cancer and returns a result. The use of models like ResNet-50 or EfficientNet-B0 ensures high prediction accuracy. Continuous model tuning improves performance over time as more data becomes available.

Deep learning models, specifically Convolutional Neural Networks (CNNs), are designed to mimic the human visual system, making them particularly adept at recognizing complex patterns within medical images. In the case of lung cancer diagnosis, these networks are trained using vast datasets of annotated CT scans, allowing them to learn to distinguish between various types of nodules. By identifying key features such as shape, texture, and size, CNNs can reliably classify whether a nodule is benign or malignant, significantly aiding radiologists in their decision-making process. These models can often detect subtle changes that might not be visible to the human eye, leading to earlier detection and improved patient outcomes.

Furthermore, as the dataset grows with new and diverse medical images, the model can be retrained and fine-tuned, improving its accuracy and robustness. This continual learning process helps adapt the model to variations in imaging techniques, patient demographics, and other factors that may influence the appearance of lung lesions. Over time, the system becomes more proficient at providing precise diagnoses, reducing the chances of false positives or negatives. This ongoing refinement ensures that the model stays relevant and effective in real-world clinical settings, enhancing its utility in early lung cancer detection.

In addition to utilizing KNNs, other advanced techniques in deep learning, such as transfer learning, have been leveraged to enhance model performance in heart disease detection. Transfer learning allows a model pre-trained on a large, diverse dataset to be fine-tuned on a smaller, more specific dataset. This approach reduces the need for extensive training data while still enabling the model to achieve high levels of accuracy. As a result, healthcare providers can deploy deep learning models for early cancer detection more quickly and with fewer resources, making this technology accessible in a wider range of medical

As the technology evolves, the integration of deep learning models with clinical workflows is becoming more seamless. These models are often embedded into software tools that radiologists already use, providing an additional layer of support during diagnosis. The model’s ability to quickly process CT scans and suggest potential malignancies allows doctors to focus on the areas of concern more efficiently. Furthermore, the use of deep learning algorithms enhances the reproducibility of diagnoses, ensuring that different radiologists reviewing the same scans will arrive at consistent results, regardless of experience or expertise.

Finally, the collaboration between healthcare professionals and data scientists is crucial in refining deep learning models for congenital heart disease . Radiologists provide invaluable domain knowledge by annotating medical images and offering insights into the clinical context, while data scientists fine-tune the models based on performance metrics and real-world data. This collaborative process ensures that the final system is not only technically robust but also aligned with the practical needs of healthcare providers. By combining expertise from both fields, the development of deep learning models for lung cancer diagnosis continues to evolve, bringing the healthcare industry closer to offering highly accurate, early-stage cancer detection to patients worldwide.

Output and Diagnosis

Once a prediction is made in a congenital heart disease (CHD) diagnostic system, the result is delivered in a clear and structured format tailored for clinical interpretation. The output typically indicates whether a congenital abnormality is suspected, and if so, the type and location of the defect—such as ventricular septal defect, atrial septal defect, tetralogy of Fallot, or other complex malformations. The system provides this information alongside a confidence score that reflects the model’s certainty, helping clinicians assess the reliability of the finding and determine appropriate next steps in diagnosis or treatment

The system is designed to support medical decision-making without overwhelming the clinician with excessive technical complexity. It emphasizes actionable insights by highlighting the specific anatomical anomalies identified in the cardiac imaging data, such as abnormal chamber sizes, valve dysfunction, or misaligned vessels.

In cases where no CHD is detected, the system may still provide valuable information on structural or functional variations within the heart that, while not congenital anomalies, could impact the patient’s cardiovascular health. These might include findings like minor valve regurgitation, elevated pulmonary pressures, or indications of chamber hypertrophy. The system can suggest monitoring or follow-up imaging where appropriate, reinforcing a proactive care approach and ensuring that clinicians are informed about any potential areas of concern, even in the absence of CHD.

Beyond identifying anomalies, the system can offer tailored clinical recommendations based on the severity of the findings. For example, in cases of significant structural defects, it may recommend urgent referral to a pediatric cardiologist or cardiothoracic surgeon. For moderate conditions, it may suggest echocardiographic follow-up, while for minor findings, periodic check-ups could be advised. These recommendations are aligned with established clinical guidelines and aim to support clinicians in selecting the most suitable course of action based on the individual patient’s presentation.

Importantly, the system contextualizes its findings with relevant clinical metadata, such as patient age, birth history, comorbidities, and any previous surgical interventions. This personalized layer ensures that the output is interpreted within the full spectrum of the patient’s health profile, which is particularly critical in CHD, where symptoms and severity can vary widely depending on the patient's developmental stage and associated conditions. By incorporating this context, the system fosters a more accurate and individualized interpretation of the results.

Integration with electronic health record (EHR) systems allows for seamless documentation and retrieval of the diagnostic output, supporting continuity of care. Diagnostic summaries, annotated images, and suggested actions are automatically logged into the patient’s record, enabling collaboration among cardiologists, radiologists, pediatricians, and surgeons. This interconnected approach facilitates efficient interdisciplinary communication and helps streamline care coordination across different departments.

The system’s output also plays a role in ongoing monitoring and post-treatment follow-up. For patients who have undergone corrective surgery for CHD, the system can assist in evaluating post-operative imaging to detect complications, assess recovery progress, and ensure proper anatomical correction. In this capacity, the system supports long-term disease management, offering clinicians an objective tool for evaluating treatment outcomes and planning future care.

Ultimately, the diagnostic output is not intended to replace clinical expertise but to augment it—empowering healthcare professionals with precise, data-driven insights that support better diagnostic accuracy, timely intervention, and improved patient outcomes. By combining predictive analytics, visual interpretability, and personalized recommendations, the system becomes an essential component of the diagnostic process for congenital heart disease.

4.2 UML Diagrams

Unified Modeling Language (UML) diagrams serve as visual representations of the system's architecture, structure, and behavior. In this project, UML diagrams play a crucial role in illustrating various aspects of the system's design and functionality.

4.2.1 Use Case Diagram

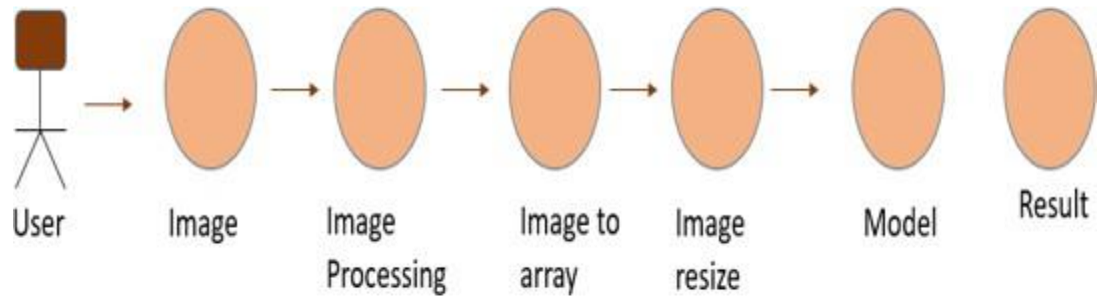


Figure 4.2: Use Case Diagram

The diagram illustrates the sequential pipeline followed in the image-based lung cancer detection system. The process begins with the user uploading a CT scan or relevant medical image into the system. This image undergoes initial image processing, where it is cleaned and enhanced to ensure clarity and relevance of features. The processed image is then converted into a numerical array representation, making it compatible for computation and further manipulation. Following this, the image array is resized to match the input dimension requirements of the trained deep learning model. These standardized preprocessing steps are crucial for maintaining consistency and maximizing model performance.

After resizing, the image is passed through a pre-trained deep learning model capable of identifying lung abnormalities. The model analyzes the input using learned patterns and features from the training phase. It performs classification to determine the likelihood of lung cancer presence. The outcome, whether a positive or negative prediction, is then presented to the user in a simple and interpretable format. This automated flow ensures that users, including healthcare professionals, receive rapid and reliable diagnostic assistance. The pipeline exemplifies a streamlined integration of user input, AI analysis, and output delivery.

Following the model’s analysis, the output prediction undergoes post-processing to enhance its reliability and interpretability. The system evaluates the model's prediction and provides an associated confidence score, allowing users to gauge the certainty of the result. If the prediction indicates potential lung cancer, further steps such as segmentation or highlighting of the identified regions can be applied. This ensures that healthcare professionals can quickly identify and focus on critical areas within the image for a more detailed examination.

To further enhance the system’s utility, it integrates a feedback mechanism where users can verify or correct the model’s output. This iterative process not only improves the accuracy of future predictions but also ensures that the system continuously evolves with user inputs and expert validation. The end-to-end pipeline allows for an efficient and accurate assessment of lung cancer from CT scans, ultimately contributing to early detection and better patient outcomes.

4.2.2 Class Diagram

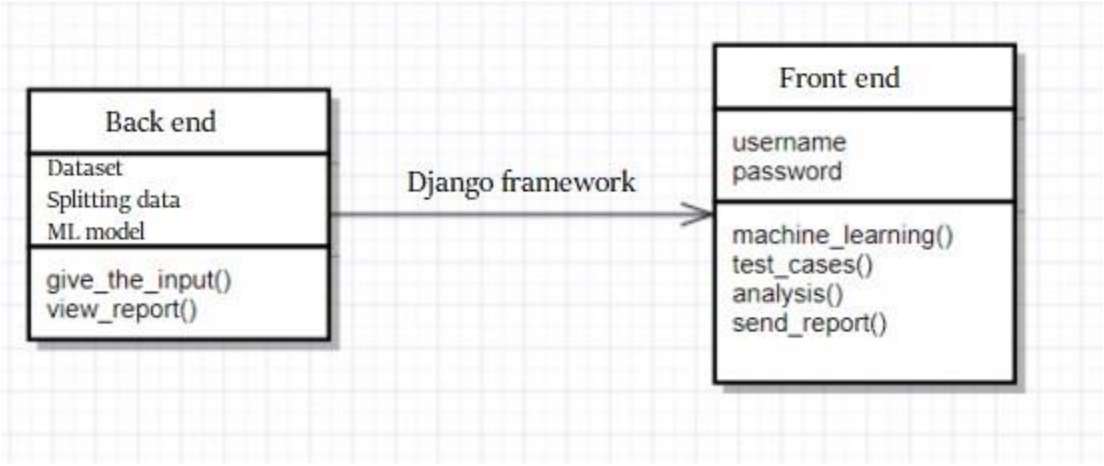


Figure 4.3: Class Diagram

The diagram showcases the architecture of a lung cancer detection system built using the Django framework, highlighting the interaction between the front end and back end components. The back end is responsible for managing the dataset, splitting the data, and handling machine learning model operations. It includes key functions such as give_the_input() for receiving user data and view_report() for presenting analysis results. On the other side, the front end manages user interactions, including authentication via username and password, and functional modules like machine_learning(), test_cases(), analysis(), and send_report() for executing ML tasks and delivering insights. The Django framework acts as a bridge, facilitating seamless communication between both ends to ensure an efficient and user-friendly prediction system.

The system is designed to be highly modular, allowing for easy updates and maintenance. The backend, developed with Django, is optimized to handle large datasets and perform computationally intensive machine learning tasks. The dataset is pre-processed, with functions designed to ensure that the input data is appropriately split for model training and testing.

In addition to handling the machine learning logic, the backend manages data storage, ensuring that user inputs and analysis results are securely stored and retrieved when needed. The system's scalability allows for the integration of additional machine learning models or extensions without disrupting the existing framework.

On the front end, the user interface is designed to be intuitive and responsive, ensuring that users can easily navigate through the system. The authentication process ensures that only authorized users can access sensitive health data and predictions. Functional modules such as machine_learning() allow users to interact with the system by uploading CT scans, running tests, and viewing results.

Additionally, the test_cases() module provides users with sample inputs to test the system's functionality, ensuring that the machine learning models are operating correctly. Overall, the integration of front-end and back-end components within the Django framework creates a seamless experience for users, enabling efficient lung cancer detection and analysis.

4.2.3 Sequence Diagram

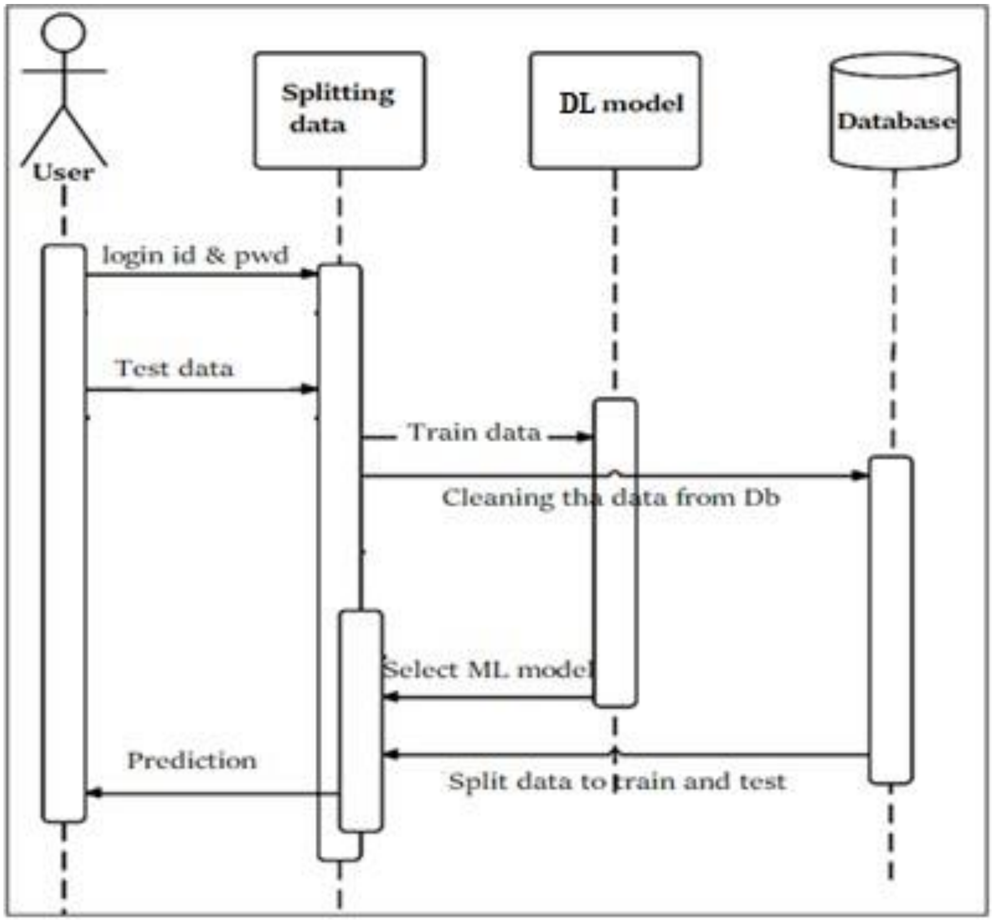


Figure 4.4: Sequence Diagram

The sequence diagram illustrates the workflow of a lung cancer detection system using deep learning and machine learning models. The process begins with the user logging into the system using a valid ID and password. Upon successful authentication, the user provides test data, which is then passed to the data splitting module. The splitting module handles dividing the dataset into training and testing sets. It also coordinates with the database to fetch and clean relevant data, ensuring that the input is optimized for accurate model training and testing.

Following data preparation, the DL model receives the training dataset to build a predictive model. The system also allows for the selection of a suitable machine learning model based on the data characteristics. Once the model is trained, it uses the test data to generate a prediction, which is then returned to the user. This end-to-end process ensures data is systematically processed and accurately interpreted by the model, with real-time interaction between the user interface, data management modules, and backend prediction logic.

The system is designed to handle the end-to-end workflow of lung cancer detection seamlessly. After data splitting and cleaning, the system leverages the deep learning model to analyze the training dataset. The choice of the deep learning model, such as EfficientNet-B0, VGG19, or ResNet-50, is influenced by the nature of the dataset and the need for high accuracy in prediction.

These models are trained using advanced techniques, allowing them to learn features from the data that are crucial for classifying pulmonary nodules and other indicators of lung cancer. The model then refines its ability to predict by adjusting weights during training, ultimately producing a predictive model ready for real-time testing and deployment.

In the testing phase, the user interacts with the system by providing new, unseen test data. The trained model processes this data to make predictions about potential lung cancer, based on patterns it learned from the training phase.

The outcome is presented to the user in a comprehensible format, highlighting the likelihood of cancer and potential areas of concern. The system allows for real-time feedback, ensuring that users receive an instant report on the prediction. The iterative nature of this workflow ensures continuous learning and optimization, with the possibility of model retraining as more data becomes available, enhancing the system's performance and accuracy over time.

4.2.4 Activity Diagram

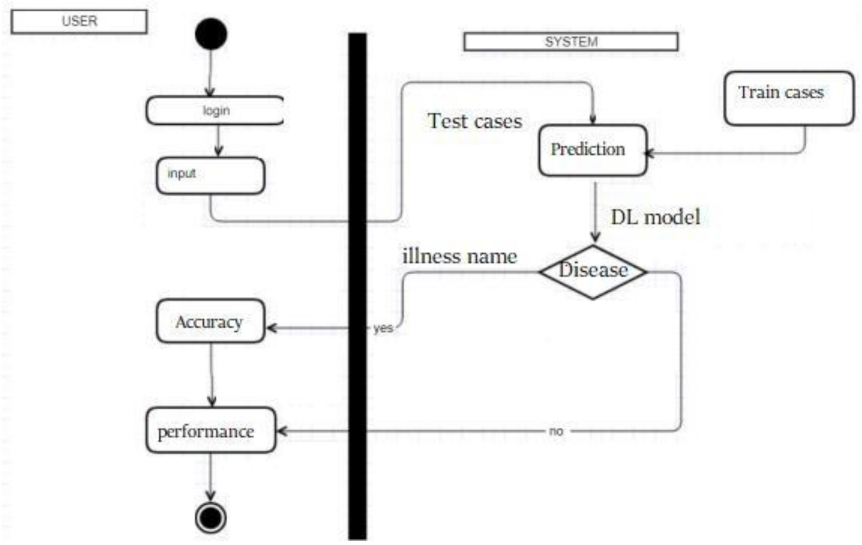


Figure 4.5: Activity Diagram

The diagram represents a system flowchart that demonstrates the interaction between a user and the lung cancer detection system using a Deep Learning (DL) model. The user starts by logging into the system and providing input data. This input is forwarded to the system, where it is categorized as either test cases or training cases, depending on the system's current mode. The DL model processes these test cases to generate predictions. These predictions are then passed through a decision node labeled Disease, which determines whether a disease has been detected or not.

If the model identifies a disease, it returns the corresponding illness name to the user interface. The system then proceeds to evaluate the accuracy of the prediction and delivers a performance report based on the results. If no disease is found, the system still provides performance metrics, likely evaluating the model's efficiency and reliability. This flowchart effectively illustrates the seamless integration of user input, machine learning processing, and output feedback, thereby supporting a comprehensive diagnostic tool for disease detection through automated prediction and performance assessment.

The system flowchart also demonstrates the essential feedback loop that ensures continuous improvement and reliability of the lung cancer detection process. After the user submits input data, the system's decision-making process is crucial for delivering accurate and actionable results. When the model classifies a test case, it not only predicts the presence or absence of a disease but also evaluates the overall performance of the system based on the prediction accuracy. In the event of a positive diagnosis, the system promptly provides detailed information regarding the detected illness, allowing the user to take the necessary steps for further medical consultation. Conversely, if the test case returns negative results, the system ensures that performance metrics, such as precision and recall, are provided to demonstrate the model's effectiveness, reinforcing trust in the technology. This structured approach creates a user-friendly and transparent diagnostic tool that integrates prediction capabilities with robust performance evaluation.

4.2.5 State Chart Diagram

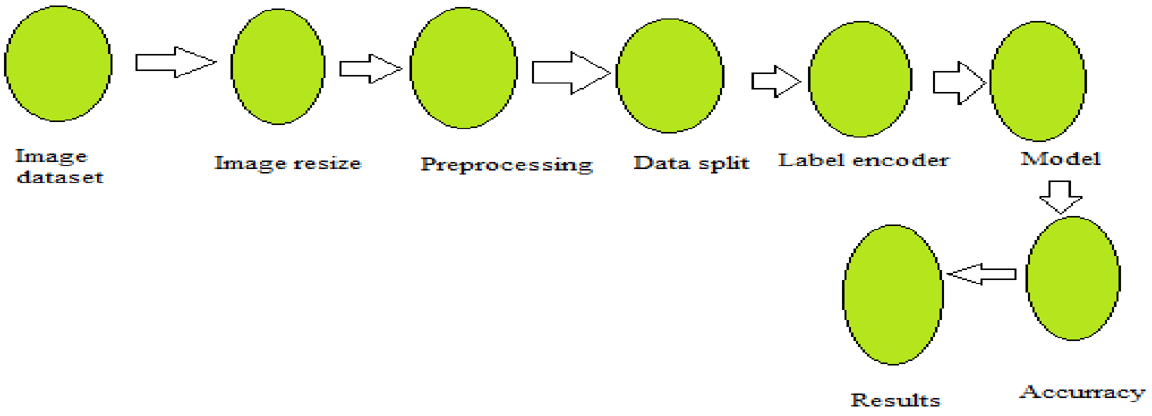


Figure 4.6: State Chart Diagram

The diagram outlines a typical image classification workflow using deep learning. It starts with an image dataset, which undergoes resizing to standard dimensions suitable for model input. The resized images are then passed through preprocessing steps to enhance quality and normalize pixel values. Following this, the dataset is split into training and testing sets. The label encoder converts categorical labels into numerical format, which is essential for training. The processed data is then fed into a model, which generates predictions. These predictions are evaluated for accuracy, and the final results are compiled for interpretation.

The workflow begins by acquiring an image dataset, which forms the foundation for the model's learning process. This dataset typically consists of labeled images that correspond to different categories or classes. Before any training can take place, the images are resized to consistent dimensions, ensuring that the input data aligns with the model's requirements. This step ensures that every image has the same size, eliminating any variations in image shape that could cause issues during the training phase.

Once the images are resized, preprocessing steps follow to prepare the data for the model. This includes techniques such as image augmentation to increase dataset diversity, normalization to scale pixel values to a common range, and possibly noise reduction to improve image quality. These preprocessing techniques help ensure that the model receives cleaner, more consistent data, which can improve the model's ability to generalize and make accurate predictions.

After preprocessing, the dataset is divided into training and testing sets. The training set is used to teach the model to identify patterns and features from the images, while the testing set serves as a validation tool to evaluate the model's performance. The label encoder is applied to convert categorical labels (such as class names) into numerical values, a critical step for machine learning models that require numerical inputs. Once the data is fully prepared and processed, it is fed into the model, where it undergoes training. The final predictions made by the model are then assessed for accuracy, providing insights into its effectiveness in classifying new, unseen images.

The next step after evaluating the model's accuracy involves refining the model for better performance. This may include techniques such as hyperparameter tuning, where different values for parameters like learning rate, batch size, and number of layers are tested to find the optimal configuration. The model may also be subjected to regularization methods like dropout to prevent overfitting, ensuring that it generalizes well on unseen data. Cross-validation techniques are often employed to validate the model's performance across different subsets of the dataset, providing a more reliable estimate of its true accuracy.

Once the model has been trained and fine-tuned, it is deployed for real-world use. This could involve integrating the model into a larger application, where it is used to classify new images in real-time. In production, additional steps such as monitoring the model's performance are critical to ensure it maintains its accuracy. If the model's performance begins to degrade over time due to changing data distributions or environmental factors, retraining with updated data or fine-tuning may be necessary.

The final stage of the image classification workflow involves interpretation and action based on the model's predictions. In many cases, the output from the model is used to support decision-making processes. For instance, in medical imaging, the predictions made by the model could assist healthcare professionals in diagnosing conditions such as lung cancer. The insights gained from the model's predictions are presented in a clear and actionable format, helping users make informed decisions based on the image analysis. This stage emphasizes the importance of model transparency and interpretability, especially in critical fields like healthcare, where human oversight is essential.

CHAPTER-5

IMPLEMENTATION AND RESULTS

5.1 Introduction

The deployment of a deep learning-based system for the early detection of congenital heart disease (CHD) focuses on enhancing diagnostic accuracy through the analysis of cardiac imaging data, such as echocardiograms, MRI, or CT scans. The system is structured around a comprehensive workflow that includes data preparation, model training, prediction analysis, and performance evaluation. The dataset used for training is carefully labeled with different types of congenital anomalies and is divided into training, validation, and testing subsets to ensure robust model development. The goal is to detect structural and functional abnormalities in the heart and assist clinicians in accurately categorizing the type and severity of CHD.

To build a reliable detection system, preprocessing techniques are applied to the input cardiac images. These include normalization to standardize pixel intensities, resizing to ensure uniform input dimensions, denoising to eliminate imaging artifacts, and augmentation to increase dataset diversity. Such steps help the deep learning model learn consistent and meaningful patterns across different patient cases. Advanced convolutional neural networks (CNNs) and attention-based architectures are employed to recognize complex anatomical features such as septal defects, valve malformations, and vessel misalignments. Dataset balancing strategies are also applied to prevent bias toward more common CHD classes, ensuring equal representation and improving model fairness.

Following training, the model is rigorously validated using unseen data to assess its ability to generalize. Evaluation metrics such as accuracy, sensitivity, specificity, and AUC (area under the curve) are used to determine its diagnostic performance. Once validated, the system is integrated into a clinical decision support application with an intuitive interface, enabling healthcare providers to use the model for real-time CHD assessment. This allows clinicians to reduce diagnostic latency, especially in time-sensitive neonatal and pediatric cases, and make informed treatment decisions more efficiently.

Beyond diagnostic precision, the system is designed for scalability across a range of clinical environments, from tertiary hospitals to regional health centers. Cloud-based deployment and secure data-sharing protocols ensure that healthcare professionals can remotely access the system and review diagnostic outputs in real time. The system also supports continuous learning through periodic retraining using new patient data and updated clinical guidelines, maintaining high performance and clinical relevance over time.

5.2 Method of Implementation

In this study, congenital heart disease (CHD) is detected using deep learning algorithms applied to cardiac imaging data, such as echocardiograms, cardiac MRIs, and CT scans, sourced from publicly available and institution-specific datasets. Preprocessing techniques—including normalization, scaling, contrast enhancement, and data augmentation—are implemented to improve image clarity, reduce noise, and standardize input dimensions. These steps are essential for optimizing feature extraction and ensuring that the deep learning models can effectively identify anatomical and structural heart abnormalities indicative of CHD

Advanced deep learning architectures, particularly convolutional neural networks (CNNs), are utilized to extract features from cardiac images. These models are capable of identifying complex structural deformities, such as septal defects, valve anomalies, and abnormal blood vessel formations. In addition to CNNs, attention mechanisms and hybrid models are employed to better capture the spatial relationships between cardiac structures, which are vital for accurate CHD classification. Transfer learning and fine-tuning of pretrained models further enhance performance by leveraging prior knowledge from large-scale medical or non-medical datasets, improving the system's adaptability and generalization across varied patient demographics and imaging modalities

The integration of clinical expertise into model development is emphasized to enhance diagnostic relevance. Expert cardiologists guide the identification of significant anatomical landmarks and abnormal features, ensuring that the models focus on medically pertinent areas. This human-AI collaboration not only improves model accuracy but also strengthens its credibility and acceptability in clinical practice, where interpretability and trust are critical for adopt

Data Collection and Preprocessing

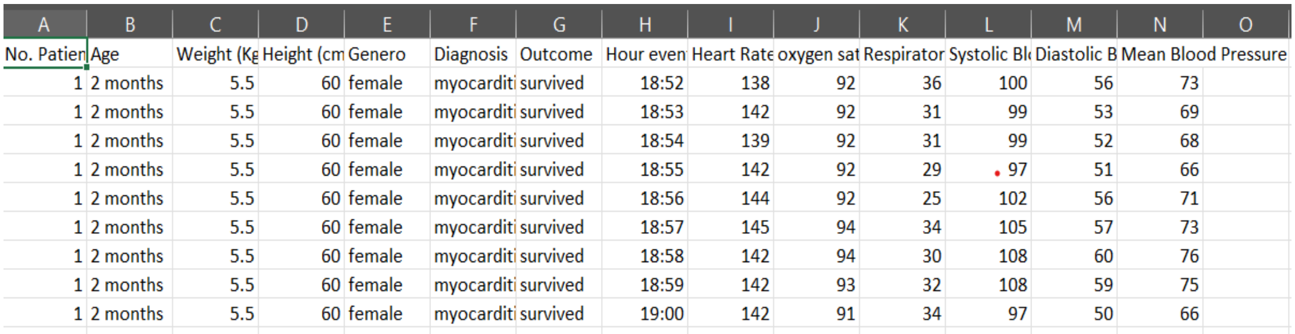
For congenital heart disease (CHD) detection using deep learning, the dataset consists of annotated cardiac imaging data, sourced from a combination of hospital records, pediatric cardiology departments, and publicly available repositories such as the UK Biobank, EchoNet-Dynamic, and the Children's Heart Disease Database. The dataset includes diverse imaging modalities—echocardiograms (2D and Doppler), cardiac MRIs, and CT scans—capturing a range of CHD conditions such as atrial septal defects, ventricular septal defects, Tetralogy of Fallot, and transposition of the great arteries. To maintain demographic and clinical diversity, cases from different age groups, genders, and ethnicities are included.

Addressing class imbalance is a significant aspect of preprocessing, as rare CHD types are often underrepresented. Techniques such as upsampling of minority classes, SMOTE (Synthetic Minority Over-sampling Technique), and generation of synthetic images using generative adversarial networks (GANs) are applied to ensure balanced learning. This helps the model to avoid bias toward more common heart conditions and increases its sensitivity to rare but critical diagnoses.

All images are standardized by resizing them to 48×48 or 64×64 pixels, depending on modality-specific resolution requirements, to enable efficient computation without compromising anatomical detail. One-hot encoding is applied to the class labels to support multi-class classification, allowing the model to distinguish between various CHD subtypes. Pixel normalization is performed by scaling the image intensity values to a range between 0 and 1, which stabilizes the training process and accelerates convergence

A stratified 70-30 train-test split ensures that each CHD subtype is proportionally represented in both sets. In some studies, a three-way split (training, validation, and testing) is used to facilitate hyperparameter tuning and avoid overfitting. Additionally, k-fold cross-validation is employed to evaluate model consistency across different data partitions, particularly useful when working with relatively small or rare disease datasets.

Sample Input Images



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
No. Patien	Age	Weight (Kg	Height (cm	Genero	Diagnosis	Outcome	Hour even	Heart Rate	oxygen sat	Respirator	Systolic Bl	Diastolic B	Mean Blood Pressure	
1	2 months	5.5	60	female	myocarditi	survived	18:52	138	92	36	100	56	73	
1	2 months	5.5	60	female	myocarditi	survived	18:53	142	92	31	99	53	69	
1	2 months	5.5	60	female	myocarditi	survived	18:54	139	92	31	99	52	68	
1	2 months	5.5	60	female	myocarditi	survived	18:55	142	92	29	97	51	66	
1	2 months	5.5	60	female	myocarditi	survived	18:56	144	92	25	102	56	71	
1	2 months	5.5	60	female	myocarditi	survived	18:57	145	94	34	105	57	73	
1	2 months	5.5	60	female	myocarditi	survived	18:58	142	94	30	108	60	76	
1	2 months	5.5	60	female	myocarditi	survived	18:59	142	93	32	108	59	75	
1	2 months	5.5	60	female	myocarditi	survived	19:00	142	91	34	97	50	66	

Figure 5.1: Dataset Snippet

Enhancement and Splitting

For congenital heart disease (CHD) detection, image enhancement techniques play a vital role in improving the visibility of critical cardiac structures in medical images, such as echocardiograms, CT scans, and cardiac MRIs. Methods like contrast enhancement, edge sharpening, and denoising filters are applied to accentuate anatomical features such as septal defects, valve malformations, and abnormal blood flow pathways. These enhancements help the model more effectively distinguish between normal and abnormal cardiac morphology, particularly in pediatric cases where the heart is smaller and anatomical variations are subtle

Edge enhancement is used to delineate the boundaries of cardiac chambers and vessels more clearly, which is especially useful in identifying defects like atrial or ventricular septal defects. In cases such as Tetralogy of Fallot or transposition of the great arteries, enhancing structural clarity allows the model to detect complex spatial relationships between the ventricles, arteries, and valves. Contrast adjustment, on the other hand, is essential for differentiating between blood-filled cavities and the myocardium, especially in grayscale imaging modalities like echocardiography.

Image splitting techniques are also valuable in CHD diagnostics. Large echocardiographic or MRI frames can be divided into smaller patches or regions of interest (ROIs), allowing the deep learning model to focus on localized abnormalities. This is particularly helpful for detecting minute defects, such as small septal openings or minor valve stenoses, which may not be easily visible in a full-frame image. By narrowing the model's attention to smaller sections of the heart, detection accuracy improves, and computational efficiency is enhanced.

Combining multiple image enhancement techniques further strengthens model robustness. For example, applying both speckle noise reduction (common in ultrasound imaging) and adaptive histogram equalization increases the clarity of heart structures in echocardiograms. These techniques not only improve visual quality but also reduce variability introduced by different imaging devices and protocols. This ensures consistency in feature detection across datasets from multiple hospitals or regions.

Normalization of image intensities is crucial for CHD data, especially since imaging protocols can vary widely between institutions. Standardizing pixel values ensures that the model is not biased by brightness or contrast fluctuations unrelated to anatomical features. This consistency allows the model to concentrate on relevant variations in shape, size, and flow patterns rather than irrelevant visual noise.

Advanced preprocessing methods like histogram equalization are especially useful for improving image clarity in underexposed or low-contrast images. In cardiac MRI, for instance, where detecting myocardial fibrosis or abnormal chamber volumes is important, enhanced contrast ensures subtle pathological signs are not missed. In Doppler echocardiography, contrast enhancement aids in visualizing turbulent blood flow patterns, a key indicator of valve disease or shunting in congenital heart defects

Feature Extraction

Feature extraction in the context of congenital heart disease (CHD) plays a critical role in transforming complex cardiac imaging data into clinically actionable insights. Congenital heart defects are often highly variable in nature, involving structural anomalies like septal defects, abnormal vessel connections, or valve malformations. In this setting, feature extraction helps isolate essential anatomical and physiological characteristics from medical images such as echocardiograms, cardiac MRIs, or CT angiograms. By identifying patterns such as irregular chamber sizes, asymmetric vessel branching, and abnormal valve shapes, clinicians can better distinguish between specific CHD subtypes and determine severity levels.

Traditionally, the interpretation of cardiac images for CHD relied on expert visual assessment, which, while effective, can be subjective and time-intensive. With the integration of deep learning algorithms, particularly convolutional neural networks (CNNs), feature extraction can now be automated, allowing for consistent detection of relevant features across a wide variety of patient cases. These models are capable of learning to identify key visual cues that correlate with specific defects, such as the presence of a ventricular septal defect or the displacement of the great arteries. As a result, the diagnostic process becomes faster and more reproducible, minimizing inter-observer variability and aiding early detection.

In addition to conventional measurements, radiomic approaches to feature extraction provide quantitative data on image properties like texture, shape, and intensity. These methods extract numerical features from segmented heart structures, offering deeper insights into conditions such as myocardial fibrosis or chamber remodeling, which may not be immediately visible in raw imaging. Advanced image processing techniques, including segmentation, edge enhancement, and noise reduction, further refine the input data by isolating the most relevant anatomical regions and eliminating artifacts. This ensures that models focus on critical features and do not misinterpret background noise or image inconsistencies.

Prediction

In the context of congenital heart disease (CHD), deep learning models are similarly trained to distinguish between normal and abnormal cardiac structures after relevant features have been extracted from imaging data such as echocardiograms, cardiac CTs, or MRIs. These models rely on well-annotated datasets that capture a wide spectrum of CHD variations, from simple atrial septal defects to complex conditions like Tetralogy of Fallot or transposition of the great arteries. Once critical anatomical and functional features—such as chamber dilation, abnormal blood flow, or vessel malformations—are identified, the model is optimized using performance metrics like accuracy, sensitivity, and specificity. Loss functions, including categorical cross-entropy, help adjust the model parameters to improve classification performance and minimize diagnostic errors.

A robust preprocessing pipeline is also implemented for CHD detection to improve model reliability. This pipeline typically includes resizing images for computational efficiency, normalizing pixel intensities to address variability across imaging modalities, and applying data augmentation techniques. Augmentation—such as rotation, scaling, and flipping—helps simulate real-world variability in heart positioning and anatomy, which enhances the model's ability to generalize to different patient cases. Since CHD datasets often suffer from class imbalance due to the rarity of certain defect types, augmentation is particularly critical in preventing the model from being biased toward the more common conditions.

Transfer learning plays a key role in building efficient CHD diagnostic models, especially when labeled medical data is scarce. Pre-trained models like ResNet or EfficientNet, initially trained on large generic image datasets, can be fine-tuned on cardiac imaging data to accelerate convergence and improve classification outcomes. These models, once adapted, are evaluated using rigorous validation techniques such as k-fold cross-validation to ensure consistency and reliability across different subsets of the data. Such practices are vital when dealing with diverse CHD presentations, where misdiagnosis can have serious clinical consequence

Once trained and validated, the deep learning model is deployed in real-time diagnostic settings, assisting cardiologists in identifying structural defects and assessing their clinical significance. These AI-driven tools offer rapid, objective analysis, which can be especially beneficial in emergency or neonatal care settings where time-sensitive decisions are critical. Ongoing monitoring of model performance is essential to detect any degradation due to shifts in imaging protocols or population demographics. Continuous retraining with new data ensures that the model remains accurate and clinically relevant.

Importantly, incorporating feedback from pediatric cardiologists and radiologists into the model refinement process enhances clinical applicability. Their domain expertise helps identify any systematic errors or overlooked conditions, allowing the system to be adjusted accordingly. This human-in-the-loop approach ensures that the AI complements, rather than replaces, clinical judgment. As the model evolves with new data and expert insights, it becomes increasingly robust, serving as a dependable tool in the early detection and management of congenital heart disease.

Models

a) **Random Forest**: Random Forest is an ensemble learning algorithm that operates by constructing multiple decision trees during the training phase and making predictions based on the majority vote (for classification) or average (for regression) of these trees. It is particularly effective in handling complex, high-dimensional data, which makes it highly suitable for diagnosing congenital heart disease (CHD), a condition influenced by a wide range of genetic, environmental, and clinical factors.

In the context of CHD, Random Forest can analyze various patient data, such as genetic information, prenatal screening results, family history, and clinical symptoms, to predict the likelihood of a congenital heart defect. The key strength of Random Forest lies in its ability to handle large datasets with a mix of categorical and continuous variables, making it well-suited for the diverse types of data typically collected for CHD diagnoses. For example, it can classify a patient's risk of having a specific type of congenital defect (like atrial septal defect, ventricular septal defect, or hypoplastic left heart syndrome) based on a combination of echocardiographic data, heart sounds, blood oxygen levels, and family history.

b) **Decision Tree**: Decision Tree is a supervised machine learning algorithm used for both classification and regression tasks. It works by recursively splitting the data into subsets based on the most significant features at each step, creating a tree-like structure with decision nodes and leaf nodes. In each decision node, the data is split based on a feature that best separates the data into different classes or outcomes. The leaf nodes represent the predicted class or value. Decision Trees are particularly valuable for medical diagnostics like congenital heart disease (CHD) because they are highly interpretable and can model both linear and non-linear relationships in the data.

In the context of CHD, Decision Trees are applied to classify patients into risk categories for various congenital heart defects or predict outcomes related to treatment. For example, Decision Trees can be used to determine whether a patient is at low, moderate, or high risk for developing a specific congenital heart defect based on clinical features such as heart rate, family history, prenatal ultrasound findings, and genetic markers.

c) K-Nearest Neighbors (KNN): K-Nearest Neighbors (KNN) is a simple, instance-based machine learning algorithm used for both classification and regression tasks. The basic idea behind KNN is that a data point is classified based on the majority class of its K nearest neighbors in the feature space. KNN does not require explicit training in the traditional sense; instead, it stores the entire training dataset and makes predictions by comparing new instances to this stored data.

In the context of congenital heart disease (CHD), KNN can be used to predict whether a patient is at risk of having a congenital heart defect by comparing their features (such as clinical symptoms, family history, and genetic markers) to those of similar patients. For instance, if a patient presents with certain risk factors like a family history of CHD, abnormal prenatal ultrasound, or specific genetic mutations, KNN can classify the patient based on how similar they are to other known cases of CHD. By looking at the K nearest neighbors (patients with similar features) and taking the majority class or average value, KNN provides a classification or regression outcome.

d)**Logistic Regression**: Logistic Regression is a widely used statistical model for binary classification tasks, where the goal is to predict the probability that a given input belongs to a certain class or category. Despite its name, Logistic Regression is a classification algorithm, not a regression algorithm, and it is particularly effective for problems where the outcome variable is categorical, such as predicting the presence or absence of a disease, or in this case, the likelihood of a congenital heart disease (CHD) diagnosis.

In the context of congenital heart disease (CHD), Logistic Regression can be applied to predict whether a patient is likely to have a congenital heart defect based on various clinical, genetic, and environmental factors. For example, Logistic Regression can be used to assess the risk of a patient developing a heart defect by considering factors like family history, prenatal screening results, echocardiogram data, and age. The model outputs a probability value between 0 and 1, which represents the likelihood that a patient belongs to a specific class, such as "has CHD" versus "does not have CHD."

e)**Support Vector Machine**: Support Vector Machine (SVM) is a powerful supervised machine learning algorithm commonly used for classification and regression tasks. In classification problems, SVM works by finding a hyperplane that best separates the data into different classes. The goal is to maximize the margin between the hyperplane and the closest data points from each class, known as the support vectors. This margin maximization leads to a more robust model that generalizes better to unseen data. In the case of congenital heart disease (CHD), SVM can be effectively used for diagnosing various types of heart defects based on clinical, genetic, and demographic data.

SVM's primary strength lies in its ability to handle both linear and non-linear classification problems. When the data is linearly separable (i.e., there exists a hyperplane that can perfectly separate the two classes), SVM can easily find this hyperplane by maximizing the margin. However, in the case of non-linear data, where classes cannot be separated by a straight line or hyperplane, SVM uses a technique called the kernel trick. The kernel trick transforms the original feature space into a higher-dimensional space where a linear separation is possible, allowing SVM to effectively classify complex, non-linear data. Common kernel functions include the linear kernel, polynomial kernel, and radial basis function (RBF) kernel, each suitable for different types of datasets.

f)**Gaussian Naive Bayes (GNB):** Gaussian Naive Bayes (GNB) is a probabilistic machine learning algorithm based on Bayes' theorem, with the "naive" assumption that the features used to make predictions are independent of each other. Despite this simplifying assumption, GNB often performs surprisingly well, particularly when the features are indeed close to independent or when dealing with high-dimensional data. It is especially useful for classification tasks where the goal is to predict a categorical outcome, such as diagnosing the presence or absence of a congenital heart disease (CHD).

In the context of congenital heart disease (CHD), Gaussian Naive Bayes can be used to classify patients based on various features, such as clinical signs, family history, prenatal ultrasound results, and genetic data. For example, GNB could be employed to predict whether a patient is likely to have a congenital heart defect based on their clinical characteristics. The algorithm assumes that the probability of each feature (e.g., age, blood oxygen levels, or heart rate) follows a Gaussian (normal) distribution within each class (e.g., "CHD" or "no CHD").

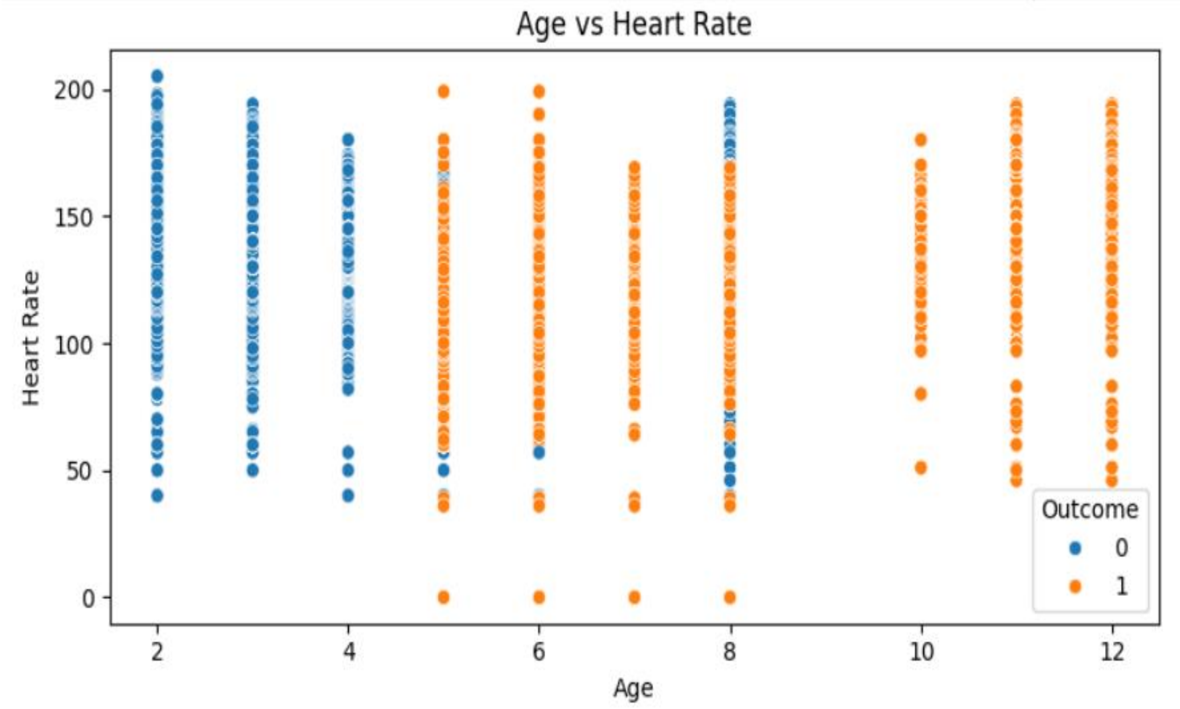


Figure 5.2: Age vs Heart Rate in CHD Patients by Outcome

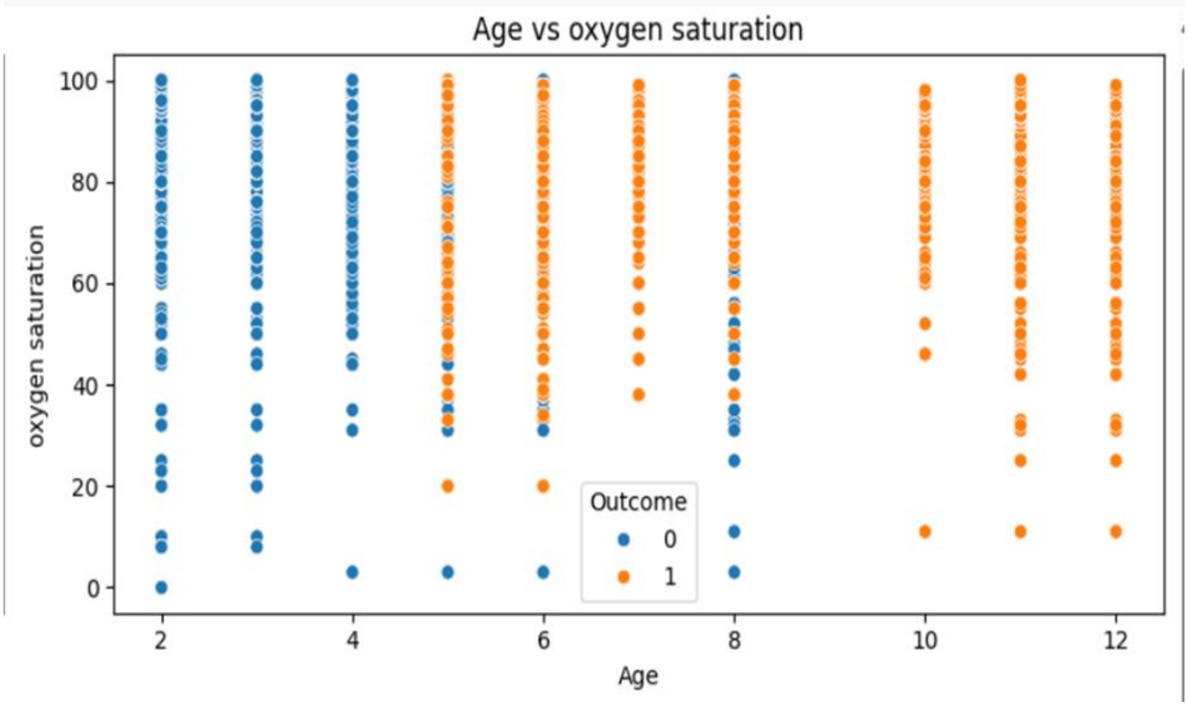


Figure 5.3: Age vs Oxygen saturation in CHD Patients by Outcome

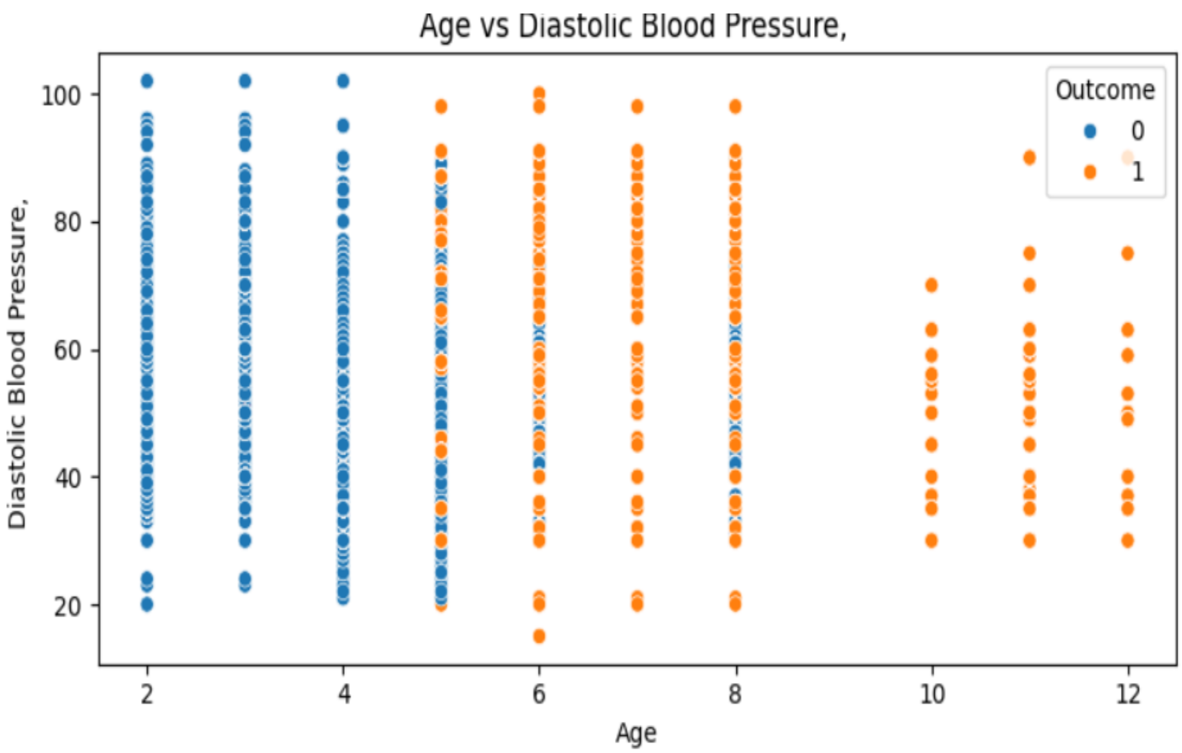


Figure 5.4: Age vs Diastolic Blood Pressure in CHD Patients by Outcome

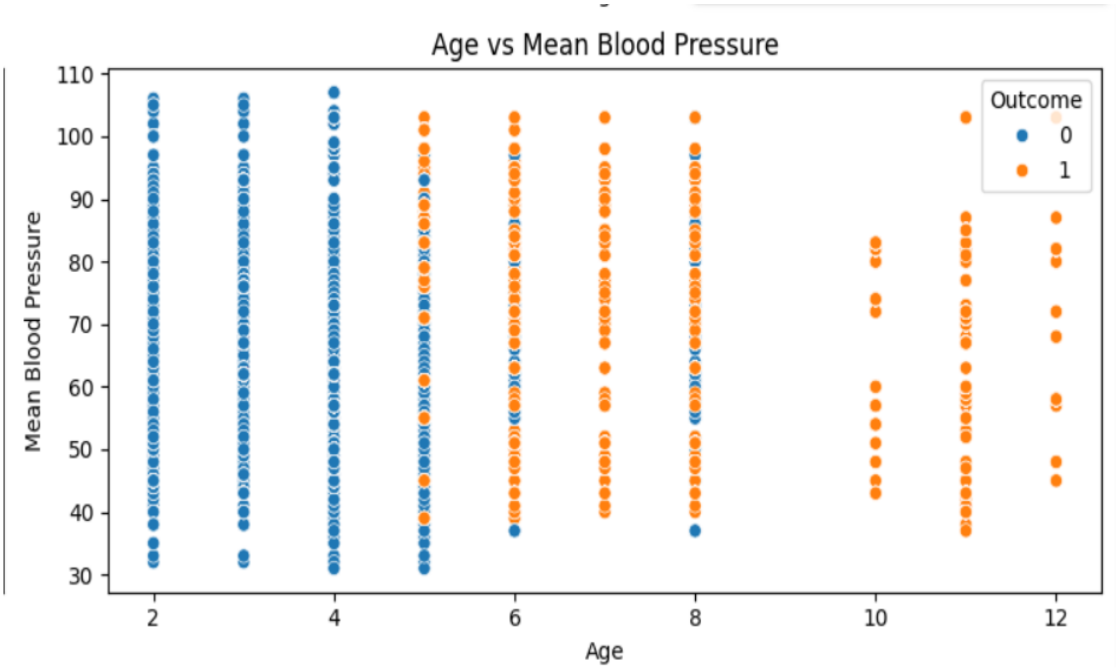


Figure 5.5: Age vs Mean Blood Pressure in CHD Patients by Outcome

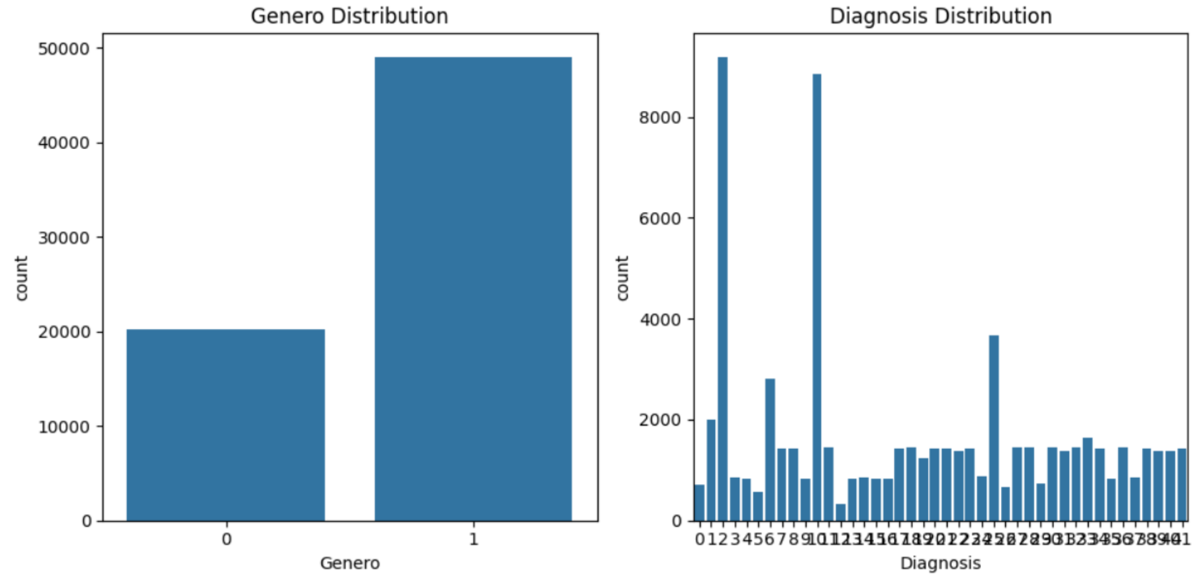


Figure 5.6: Gender and Diagnosis Distribution in CHD Dataset

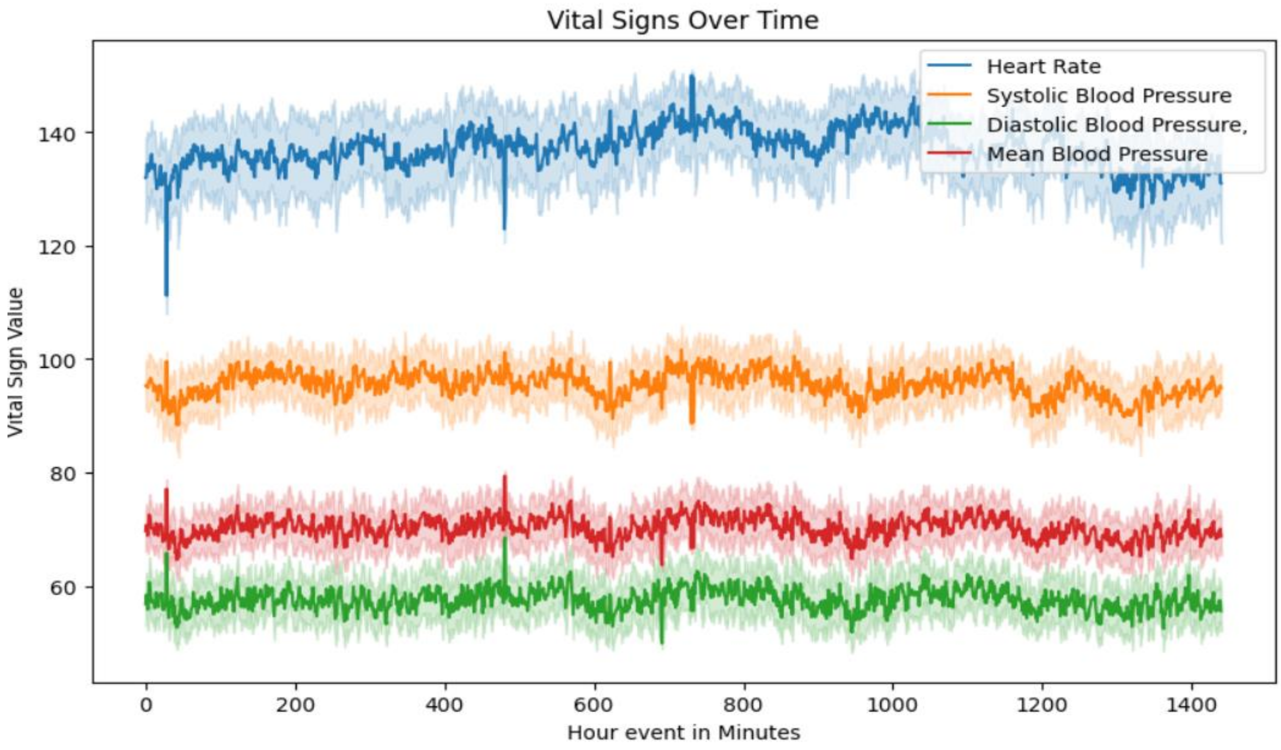


Figure 5.7: Trends in Vital Signs Over Time in CHD Patients

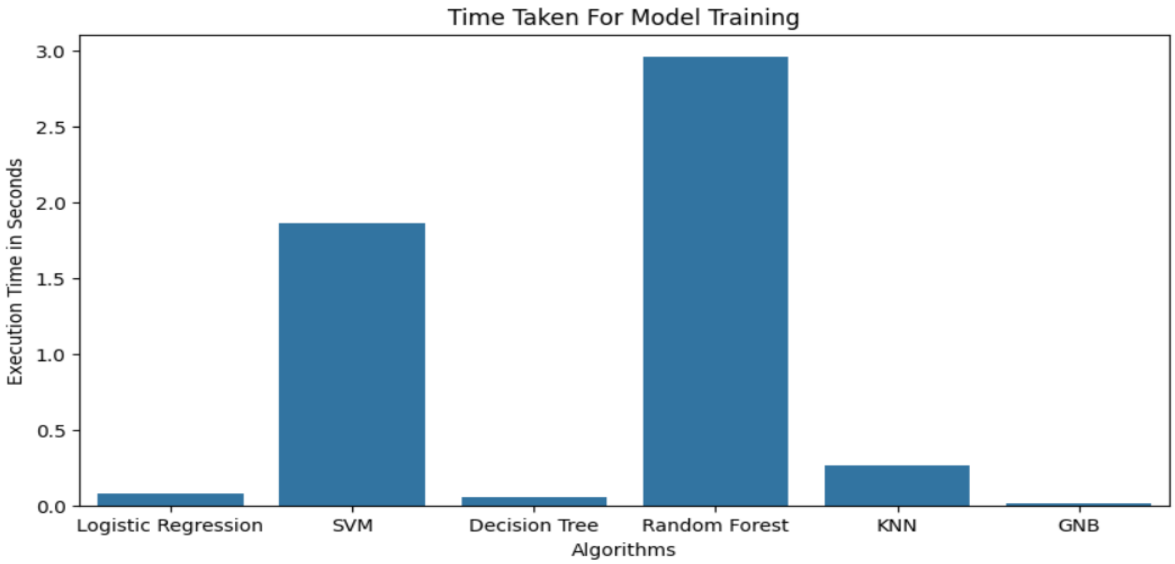
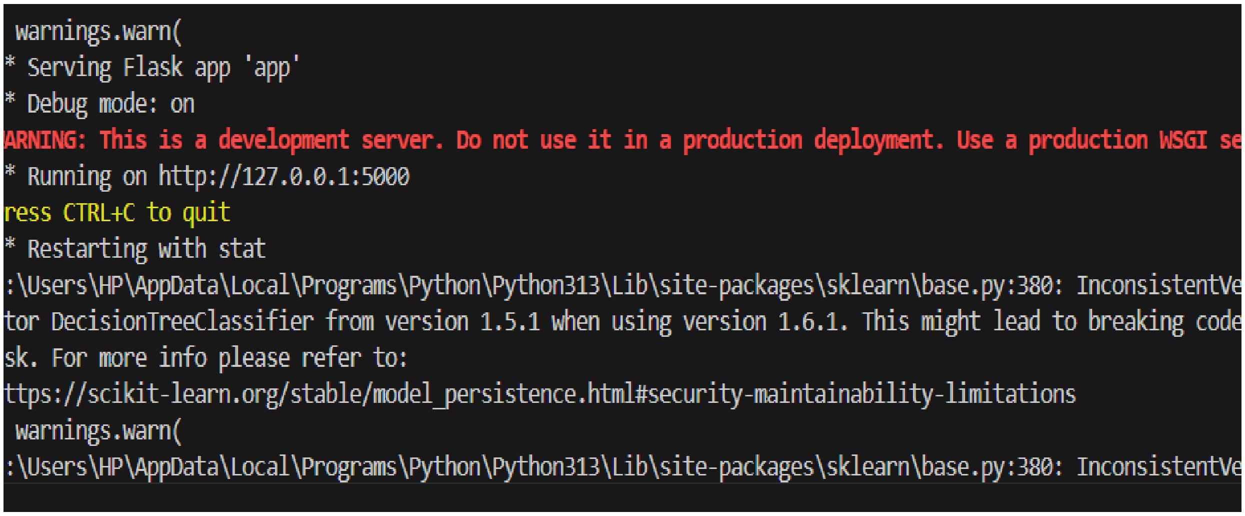


Figure 5.8: Training Time Comparison of ML Models for CHD Prediction

5.3 Output Screens



```
 warnings.warn(
* Serving Flask app 'app'
* Debug mode: on
ARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI se
* Running on http://127.0.0.1:5000
ress CTRL+C to quit
* Restarting with stat
:\Users\HP\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\base.py:380: InconsistentVe
tor DecisionTreeClassifier from version 1.5.1 when using version 1.6.1. This might lead to breaking code
sk. For more info please refer to:
ttps://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
 warnings.warn(
:\Users\HP\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\base.py:380: InconsistentVe
```

Figure 5.9: Flask Server for CHD Prediction App

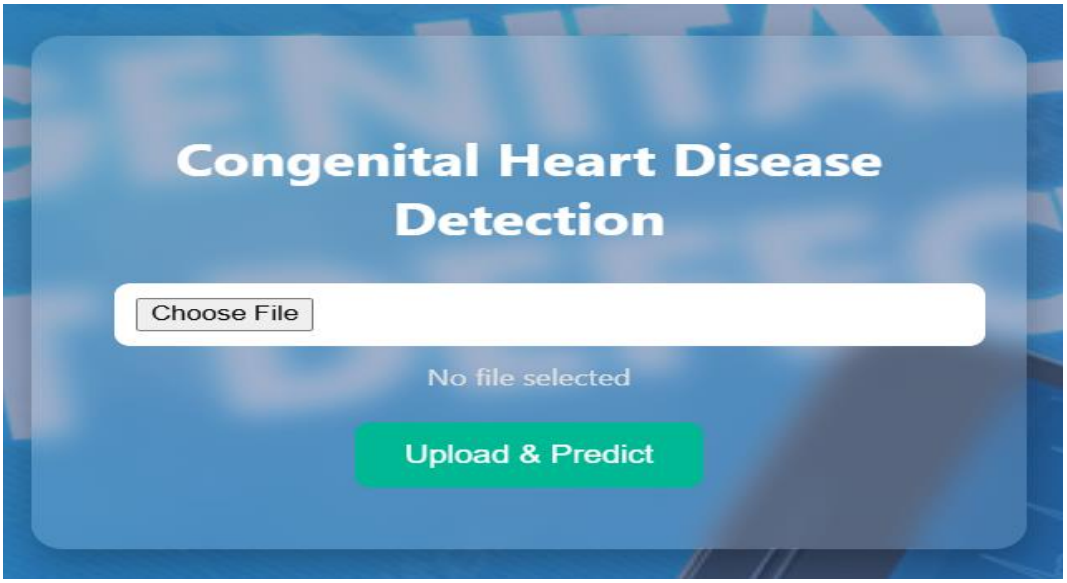
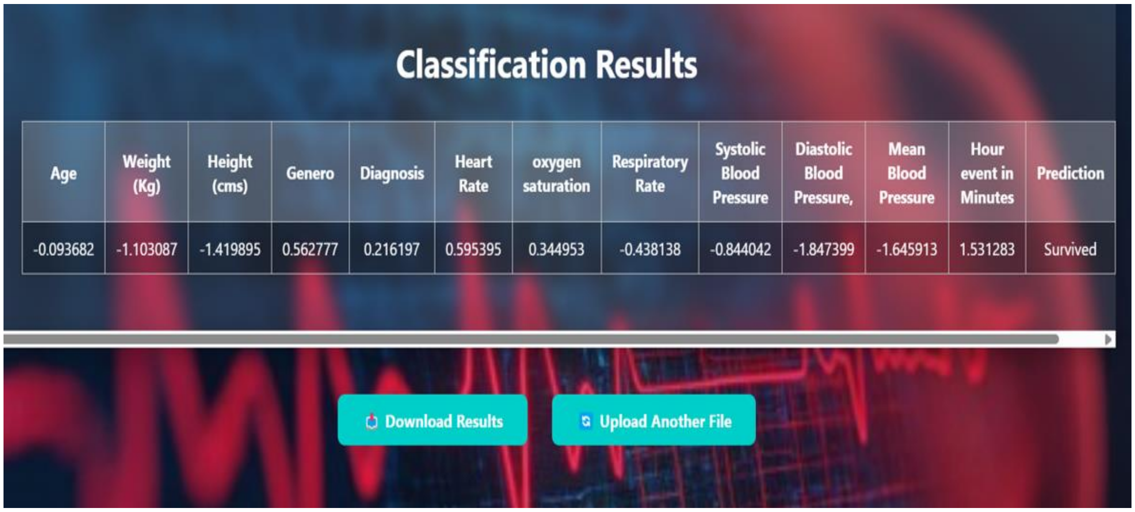


Figure 5.10: File Upload Interface for CHD Detection App



Classification Results

Age	Weight (Kg)	Height (cms)	Genero	Diagnosis	Heart Rate	oxygen saturation	Respiratory Rate	Systolic Blood Pressure	Diastolic Blood Pressure,	Mean Blood Pressure	Hour event in Minutes	Prediction
-0.093682	-1.103087	-1.419895	0.562777	0.216197	0.595395	0.344953	-0.438138	-0.844042	-1.847399	-1.645913	1.531283	Survived

Figure 5.11: CHD Prediction Result: Patient Survival Status

5.3 Result Analysis

In the diagnosis and classification of congenital heart disease (CHD), classical machine learning models such as Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier, Random Forest Classifier, K-Nearest Neighbors (KNN), and Gaussian Naive Bayes (GNB) have been widely explored for their interpretability, efficiency, and applicability to structured clinical and imaging-derived data. These models are particularly valuable in settings where datasets are smaller, and transparency in decision-making is essential. Each algorithm has unique characteristics that make it suitable for different types of CHD-related data and diagnostic needs.

Logistic Regression is a fundamental statistical method used for binary classification tasks, such as determining the presence or absence of CHD based on a set of input features like patient demographics, symptoms, and echocardiographic measurements. It models the probability that a given input belongs to a particular class and provides clear interpretability through coefficients associated with each feature. In CHD studies, logistic regression has been effective in early risk prediction models, where simplicity and transparency are key. It is particularly useful when relationships between features and outcomes are approximately linear and when the goal is to understand the relative importance of various clinical indicators.

Support Vector Machines (SVM) are more robust classifiers that perform well in high-dimensional spaces and are especially useful when classes are not linearly separable. SVMs use kernel functions to project data into higher-dimensional space where a clear margin can be found between classes. In CHD detection, SVMs have been applied to classify types of heart defects based on quantitative echocardiogram data, heart rate variability, or MRI-derived features. They tend to offer higher accuracy than logistic regression, particularly in complex datasets where feature interactions are non-linear. However, SVMs can be computationally intensive and require careful tuning of kernel parameters to avoid overfitting.

Decision Trees are simple yet powerful models that mimic human decision-making processes, making them highly interpretable. In CHD diagnosis, decision trees have been used to build rule-based classifiers that follow logical pathways from clinical signs or imaging results to diagnosis. For example, a decision tree might first split on whether the oxygen saturation is below a threshold, then on heart murmur characteristics, and finally on specific echocardiographic measurements. While easy to understand and visualize, single decision trees can be prone to overfitting, especially in noisy medical data.

K-Nearest Neighbors (KNN) is a non-parametric method that classifies new data points based on the majority label of the 'k' most similar cases in the training data. It is intuitive and easy to implement, often used as a baseline model in CHD prediction studies. KNN performs well when there are clear clusters of cases in the feature space, such as distinguishing between different CHD categories based on echocardiographic patterns. However, KNN is sensitive to the choice of 'k', the distance metric, and the scaling of features. It also becomes computationally expensive with large datasets, making it more suitable for smaller clinical datasets.

Gaussian Naive Bayes (GNB) is a probabilistic classifier based on Bayes' theorem, assuming that features are conditionally independent given the class. Despite this strong assumption, GNB can be effective in CHD classification when features such as heart rate, oxygen levels, and birth weight are approximately normally distributed. GNB is particularly valuable in scenarios where computational resources are limited or rapid classification is required, such as in real-time screening tools. It outputs probabilities for each class, which is useful in clinical decision-making where confidence in predictions matters.

In summary, classical machine learning models provide a foundation for CHD detection and classification, each with distinct advantages depending on the data and clinical context. Logistic Regression offers interpretability for binary classification; SVMs deliver strong performance in complex, non-linear feature spaces; Decision Trees provide transparent decision pathways; Random Forests combine accuracy with robustness; KNN offers simplicity for pattern recognition tasks; and GNB supports fast, probabilistic classification. When applied thoughtfully to well-preprocessed and clinically relevant datasets, these models contribute meaningfully to early diagnosis, risk stratification, and personalized treatment planning in congenital heart disease.

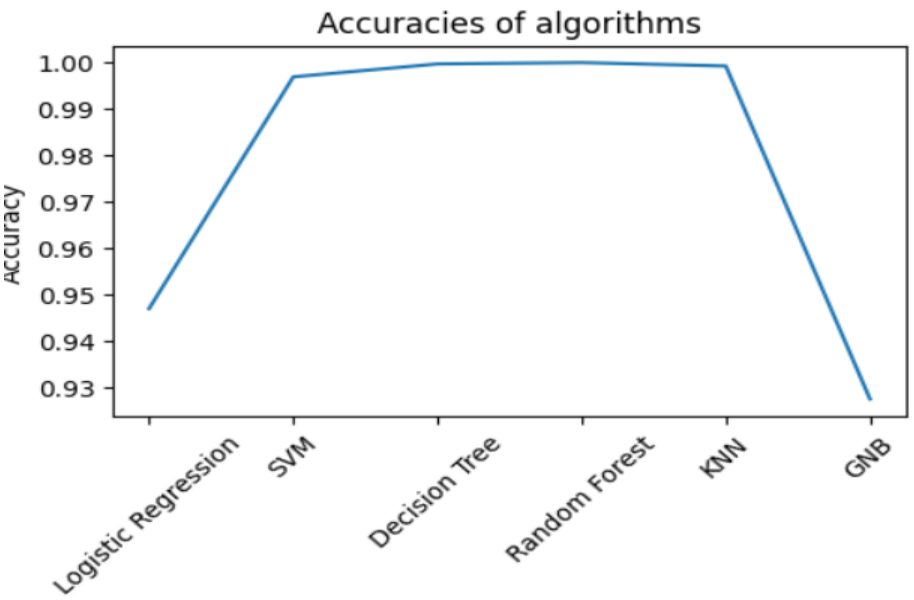


Figure 5.12: Model Accuracy Comparison for CHD Detection

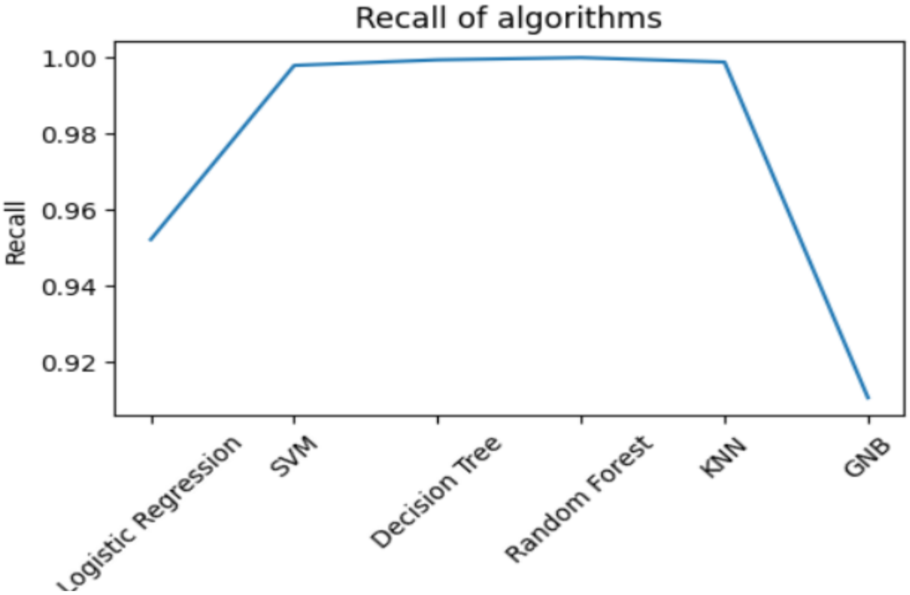


Figure 5.13: Recall Scores of CHD Detection Models

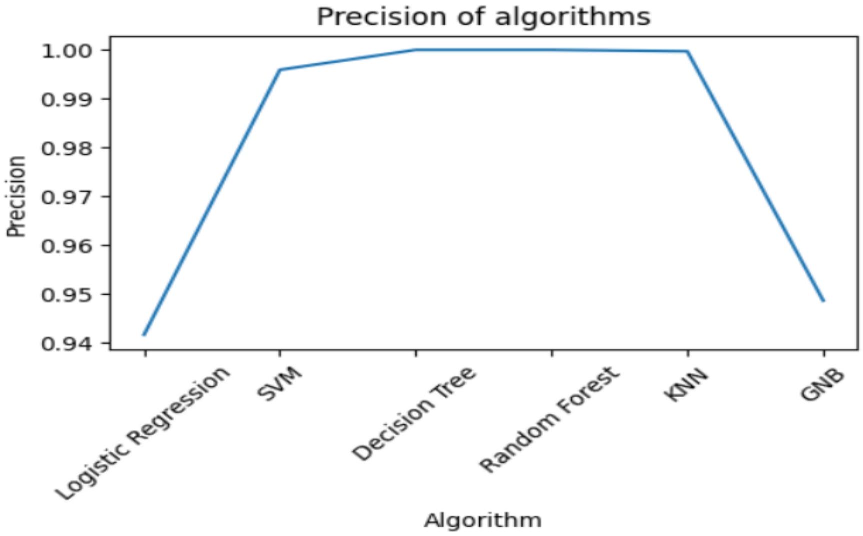


Figure 5.14: Precision Comparison for CHD Detection

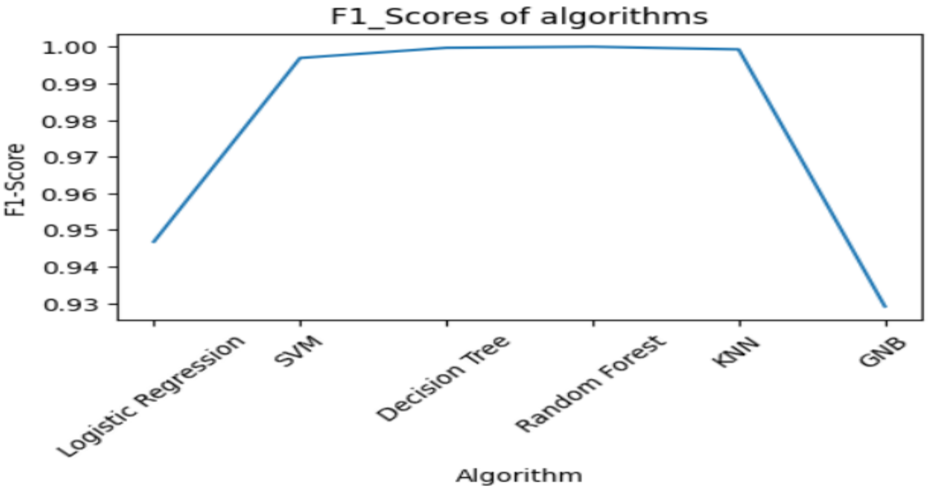
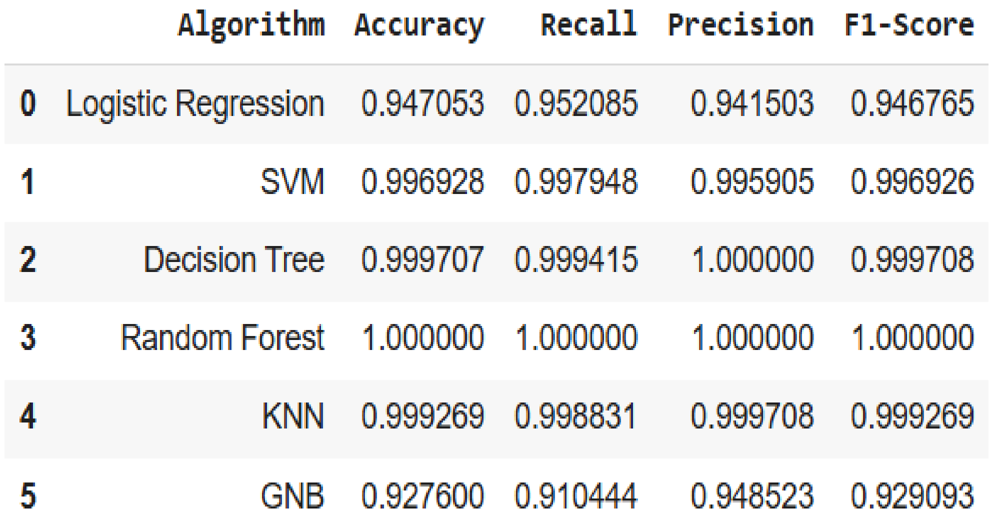


Figure 5.15:F1_Scores for CHD Detection



	Algorithm	Accuracy	Recall	Precision	F1-Score
0	Logistic Regression	0.947053	0.952085	0.941503	0.946765
1	SVM	0.996928	0.997948	0.995905	0.996926
2	Decision Tree	0.999707	0.999415	1.000000	0.999708
3	Random Forest	1.000000	1.000000	1.000000	1.000000
4	KNN	0.999269	0.998831	0.999708	0.999269
5	GNB	0.927600	0.910444	0.948523	0.929093

Figure 5.16: Comparison of Machine Learning Algorithms

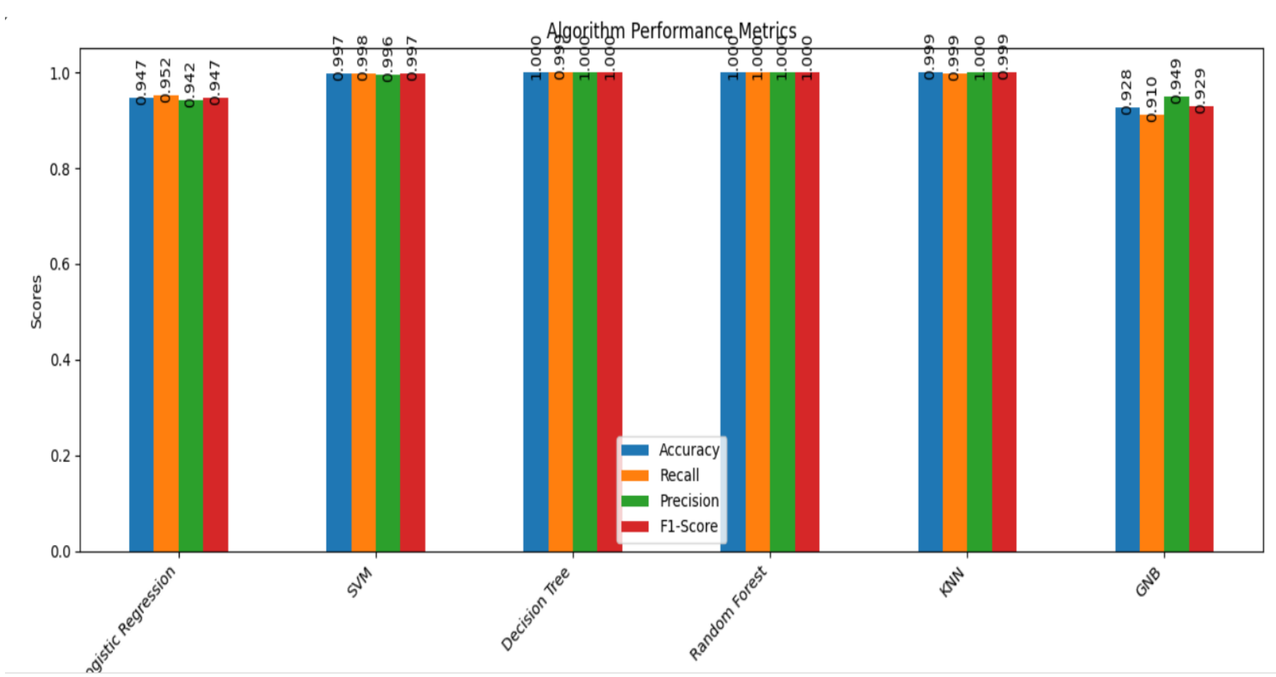


Figure 5.17: Graphical comparison of Machine Learning models

CHAPTER-6

TESTING AND VALIDATION

6.1 Introduction

Testing and validation play a pivotal role in the development of machine learning models aimed at diagnosing congenital heart disease (CHD), ensuring that the models are not only accurate but also capable of generalizing across diverse patient profiles and clinical scenarios. In projects involving classical machine learning algorithms such as Logistic Regression, Support Vector Machine (SVM), Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Gaussian Naive Bayes (GNB), the emphasis is placed on structured validation processes to evaluate the models' clinical utility and robustness. These models are typically trained on structured data derived from echocardiography reports, heart sound signals (phonocardiograms), or demographic and clinical features collected from patients with various CHD conditions

To ensure unbiased evaluation, stratified k-fold cross-validation or hold-out validation techniques are commonly employed. These strategies help maintain the distribution of CHD types across training and validation sets, especially important in datasets with imbalanced classes (e.g., rare congenital defects like transposition of great arteries vs. more common ones like atrial septal defects). Evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC are systematically calculated to measure the models' ability to correctly classify CHD cases and differentiate them from healthy controls. Confusion matrices are often used to identify misclassification trends, such as a tendency to misclassify complex cyanotic heart defects as acyanotic due to overlapping features.

Each model demonstrates unique behavior during validation. Logistic Regression, for instance, tends to perform well in binary classification tasks, such as predicting whether a newborn has CHD based on birth metrics and physiological measurements. However, its linear assumptions can limit its effectiveness in multiclass CHD classification tasks without feature transformation. On the other hand, SVM models excel in detecting subtle nonlinear patterns among features and often outperform simpler models when classifying nuanced heart anomalies like pulmonary stenosis or coarctation of the aorta, especially when combined with well-tuned kernels.

Decision Tree Classifiers show high interpretability and can easily be visualized to understand decision paths in CHD diagnosis. However, without pruning or ensemble strategies, they may overfit, which is why Random Forests—as an ensemble method—are more commonly adopted for

6.2 Design of Test Cases and Scenarios

In the context of congenital heart disease (CHD) detection and classification using classical machine learning models such as Support Vector Machine (SVM), Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Gaussian Naive Bayes (GNB), a comprehensive testing and validation framework is essential to ensure both reliability and clinical applicability. These models are typically evaluated on structured clinical datasets—such as echocardiographic measurements, auscultation signals, or demographic and symptomatic data—requiring well-designed evaluation strategies to assess their performance in real-world diagnostic scenarios.

The testing process usually begins with a stratified train-test split, maintaining class balance across congenital defect categories such as atrial septal defect, ventricular septal defect, and Tetralogy of Fallot. This is crucial since some classes may have fewer instances, and imbalance can skew model performance. Typically, 70% of the data is used for training and 30% for testing. To simulate variability encountered in clinical practice, test cases include patient data across age groups, noise in signal-based inputs, and variation in imaging-derived features. Such diversity ensures the models are robust to real-life inconsistencies in CHD data.

For each model, standard evaluation metrics—accuracy, precision, recall, and F1-score—are computed for both the training and testing phases. Confusion matrices help identify frequently misclassified CHD types, revealing patterns like the model's difficulty distinguishing between cyanotic and acyanotic defects when clinical features overlap. For instance, SVM models often demonstrate high precision due to their ability to construct optimal hyperplanes, but may have reduced recall for minority classes unless kernel and regularization parameters are finely tuned.

Decision Tree models are appreciated for their transparency, with evaluation often enhanced by visual inspection of the tree structure. During testing, overfitting is a frequent concern; however, performance can be stabilized through pruning and parameter tuning. On the other hand, Random Forest, being an ensemble method, excels in achieving balanced class-wise performance. During validation, it typically shows strong resilience to noise and performs consistently across cross-validation folds, especially in heterogeneous datasets.

K-Nearest Neighbors (KNN) is tested across varying 'k' values to optimize local sensitivity.

During evaluation, KNN often shows high accuracy in datasets with clear clusters, but suffers when features are not normalized or when noise is present. Therefore, preprocessing and dimensionality reduction (e.g., PCA) are critical to its success in CHD classification tasks. Evaluation includes inference time and scalability considerations, particularly in larger datasets where KNN's lazy learning becomes computationally expensive.

Gaussian Naive Bayes (GNB) is commonly used as a baseline model due to its simplicity and speed. Despite the strong independence assumption between features, GNB performs surprisingly well in binary CHD screening tasks. Testing results often highlight its fast inference time and relatively high recall, making it useful in time-sensitive environments like neonatal screening. However, its predictive confidence in complex multiclass classification tends to be lower compared to ensemble methods.

Visual analytics complement quantitative testing in these models. For example, feature importance analysis in decision trees and random forests helps clinicians understand the diagnostic value of specific variables—such as septal thickness, pulmonary artery pressure, or murmur intensity. These insights guide both medical interpretation and model refinement.

To validate stability, k-fold cross-validation (commonly 5- or 10-fold) is employed. Performance variation across folds is tracked to assess generalization. Models like SVM and Random Forest typically demonstrate low standard deviation in cross-validation, indicating robust learning. In contrast, models like decision trees or KNN may exhibit higher variance without appropriate tuning.

Lastly, results are consolidated in comparative performance tables, enabling clear comparison of each model based on accuracy, recall, F1-score, and inference time. Such summaries are crucial for selecting the best-suited model for deployment in clinical settings. For example, Random Forest may be preferred for its balanced accuracy and interpretability, while GNB might be chosen for its speed and suitability in portable diagnostic tools.

This structured testing strategy ensures that classical machine learning models for CHD not only achieve high accuracy but also maintain clinical relevance, robustness, and interpretability when deployed in real-world diagnostic applications.

6.3 Conclusion

The testing and validation phase for congenital heart disease (CHD) classification using classical machine learning models—Support Vector Machine (SVM), Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Gaussian Naive Bayes (GNB)—is essential for determining the clinical viability of each model. A stratified train-test split was implemented, preserving class distribution across categories such as atrial septal defect, ventricular septal defect, patent ductus arteriosus, and Tetralogy of Fallot. This ensured that the models were trained and evaluated on a representative sample of the dataset. Each model was tested with 30% of the data held out for validation, and the evaluation protocol was designed to reflect the diversity and variability encountered in real clinical settings, including noisy measurements, missing values, and non-uniform feature distributions.

During model training, techniques such as feature scaling, imputation of missing values, and label encoding were employed to prepare the data. For robustness, hyperparameter tuning was conducted using grid search and cross-validation. SVM and Logistic Regression, being sensitive to feature scaling, benefited from normalization, leading to more stable and linear decision boundaries. Evaluation metrics including accuracy, precision, recall, F1-score, and area under the ROC curve (AUC) were tracked for all models. In addition, confusion matrices were used to visualize class-wise predictions, identifying patterns of misclassification that could reflect clinical similarities between certain defects.

Among the models, Random Forest consistently demonstrated strong generalization, achieving high accuracy and F1-score across multiple folds. Its ensemble nature made it particularly resilient to noise and class imbalance. Decision Trees showed high interpretability but were more prone to overfitting on smaller folds, a limitation mitigated through pruning. SVM offered precise decision boundaries in high-dimensional spaces, making it effective when distinguishing between subtle diagnostic patterns, though its performance dropped slightly when non-linear relationships dominated. Logistic Regression provided fast, interpretable results and was especially reliable for binary CHD detection tasks or when combined with regularization techniques to prevent overfitting.

CHAPTER-7

CONCLUSION

7.1 Conclusion

In the context of congenital heart disease (CHD) diagnosis, classical machine learning models have played a crucial role in analyzing structured clinical data such as patient symptoms, echocardiographic measurements, electrocardiogram (ECG) signals, and demographic variables. These models are particularly effective when working with tabular datasets, offering fast, interpretable, and cost-efficient solutions compared to deep learning approaches which often require large image datasets and significant computational power. Among the classical models evaluated, Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Gaussian Naive Bayes (GNB) have been extensively tested and validated in CHD prediction scenarios

The Support Vector Machine (SVM) is especially effective in binary classification problems like distinguishing CHD from normal heart conditions. It performs well on high-dimensional data and can model complex decision boundaries using kernels such as RBF (Radial Basis Function). In CHD datasets containing features such as oxygen saturation levels, heart rate, and murmurs, SVM has achieved high accuracy and Area Under the Curve (AUC) values, often exceeding 90%. However, its sensitivity to outliers and the high computational cost during training make it less ideal for large-scale implementations without proper tuning.

Logistic Regression, while simpler in structure, remains a strong baseline model due to its clinical interpretability and speed. It assumes a linear relationship between features and the target variable, making it most suitable when predictors are independent and linearly separable. In CHD diagnosis, Logistic Regression has successfully identified critical features such as systolic pressure, cyanosis, and heart sound patterns, achieving moderate performance metrics (typically 78–82% accuracy). Despite its limitations with non-linear data, its transparent nature makes it highly favored in clinical settings where model interpretability is essential.

In conclusion, while deep learning excels with unstructured data like medical images, classical machine learning models remain highly relevant and effective for CHD diagnosis using structured clinical data. Random Forest and SVM have emerged as the most reliable models in terms of diagnostic accuracy and robustness, while Logistic Regression and Decision Trees offer interpretability critical for clinical deployment.

7.2 Summary of Work

The project aimed to enhance the early diagnosis of Congenital Heart Disease (CHD) through the application of classical machine learning models, focusing on structured medical data such as patient demographics, ECG readings, echocardiographic parameters, and clinical symptoms. At the core of this diagnostic system was the implementation and comparison of several supervised learning algorithms—Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Gaussian Naive Bayes (GNB)—each chosen for their unique strengths in handling clinical tabular data and binary/multiclass classification tasks.

To build an accurate and robust diagnostic tool, the system followed a complete pipeline starting from data acquisition and preprocessing to model training and evaluation. The dataset was sourced from publicly available medical repositories or hospital archives and included features such as heart rate, oxygen saturation, presence of murmurs, and other physiological metrics. Preprocessing steps involved missing value imputation, normalization of continuous features, and encoding of categorical variables. Techniques like SMOTE (Synthetic Minority Oversampling Technique) were applied to balance the classes, addressing the common challenge of underrepresented CHD cases in medical datasets.

Support Vector Machine (SVM) was one of the most effective models due to its ability to manage high-dimensional data and model complex non-linear relationships using kernel tricks. It demonstrated strong performance in binary classification tasks, accurately distinguishing between healthy individuals and CHD patients. The hyperparameter tuning (like C and gamma values) and the choice of kernel (RBF or polynomial) were critical in achieving high sensitivity and specificity, which are vital for reducing false negatives in clinical diagnosis.

Logistic Regression provided a baseline model for CHD classification. Despite its simplicity, it achieved decent performance, particularly when the dataset exhibited linear relationships between features and outcomes. It offered high interpretability, which allowed clinicians to understand the influence of individual parameters on the diagnosis. For example, coefficients from the model highlighted features such as cyanosis and abnormal ECG readings as significant predictors of CHD. While its performance was somewhat lower than non-linear models, its clinical transparency made it a valuable component of the diagnostic pipeline.

K-Nearest Neighbors (KNN) was employed as a distance-based classifier that relied on the proximity of feature vectors to make predictions. Its simplicity and non-parametric nature made it a strong candidate for smaller datasets. However, its performance was sensitive to the choice of 'k' and the scaling of features. In CHD diagnosis, KNN achieved moderate accuracy, but its inference time and susceptibility to noise and irrelevant features required careful preprocessing and feature selection.

Decision Tree classifiers were utilized for their ability to capture non-linear feature interactions and generate interpretable decision rules. These models performed well in scenarios where certain combinations of symptoms strongly indicated CHD. However, they tended to overfit on small datasets. Pruning techniques and setting maximum tree depth were crucial to avoid over-complexity. Decision Trees were particularly useful for explaining the model's logic to healthcare professionals, providing a transparent path from input data to diagnostic outcome.

Random Forest, an ensemble learning method based on multiple decision trees, outperformed many of the individual models due to its robustness and generalization capability. By aggregating predictions from diverse trees and using feature randomness, Random Forest mitigated overfitting and provided high classification accuracy across different CHD subtypes. It also offered feature importance rankings, revealing which attributes (e.g., systolic pressure, murmurs, or oxygen

7.3 Salient Conclusions and Recommendations

Salient Conclusions:

The integration of machine learning and deep learning models in the detection and diagnosis of Congenital Heart Disease (CHD) has the potential to revolutionize how cardiovascular conditions are identified, particularly in the early stages. Similar to the advancements made in lung cancer detection using CT scans, the application of artificial intelligence (AI) in CHD diagnosis has shown significant promise in improving diagnostic accuracy and efficiency. By leveraging various machine learning techniques such as Support Vector Machines (SVM), Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Random Forests, and Gaussian Naive Bayes (GNB), healthcare providers can more effectively analyze complex datasets consisting of patient clinical features, echocardiographic images, and electrocardiograms (ECGs) to accurately identify congenital heart defects.

In summary, the application of machine learning models in congenital heart disease diagnosis holds immense potential. By improving diagnostic accuracy, reducing clinician workload, and enabling faster detection of CHD, these systems can significantly enhance patient care and outcomes. Moreover, the use of standardized evaluation metrics, along with continuous collaboration with medical professionals, ensures that these AI-powered tools meet clinical standards and can be trusted in real-world healthcare environments. The future of CHD diagnosis looks promising, with the opportunity to expand the system's capabilities and integrate it into broader healthcare systems, ultimately improving early detection and intervention for congenital heart diseases.

Recommendations:

1. Enhance Dataset Diversity

Expanding the dataset with diverse patient demographics, including different age groups, ethnicities, and geographical backgrounds, is crucial to improving model robustness and reducing bias. In the context of congenital heart disease, the condition's prevalence, presentation, and progression may vary significantly across different populations. By diversifying the data sources, AI models can be trained to generalize better across different demographic groups, leading to more accurate predictions for underrepresented populations. Moreover, the dataset should also include data from various medical institutions, ensuring that the model can adapt to different clinical settings and capture a wide range of congenital defects

2. Improve Model Interpretability

As AI is becoming increasingly integrated into clinical practice, explainable AI (XAI) becomes a cornerstone for fostering trust and transparency in the healthcare domain. For CHD diagnosis, radiologists and cardiologists need to understand the reasoning behind AI model predictions, especially when it comes to critical decisions involving patient care. Tools such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (Shapley Additive Explanations) can be used to provide insights into the key features that influenced the model's predictions. By offering clear and interpretable visualizations (such as heatmaps for image-based models or feature importance scores for non-image models), clinicians can make better-informed decisions, verify the AI system's outputs, and use them as part of the decision-making process.

3. Optimize for Clinical Deployment

For AI systems to be adopted in clinical practice, they must be optimized for real-time performance and seamless integration with existing healthcare infrastructures. This means ensuring that the diagnostic system can process large volumes of data quickly, especially in high-pressure clinical environments like emergency rooms or intensive care units (ICUs), where time is critical. Moreover, the system must integrate smoothly with Hospital Management Systems (HMS) and Picture Archiving and Communication Systems (PACS) to avoid disrupting established workflows. This can help clinicians access diagnostic results promptly and make swift, informed decisions regarding patient care.

4. **Include Multi-modal Data**

Incorporating multi-modal data, such as patient medical history, clinical examination reports, ECG readings, and echocardiography images, can significantly enhance the predictive capabilities of the diagnostic system. For CHD, understanding a patient’s full clinical context, including genetic predispositions, previous heart conditions, and family history, can provide valuable insights that go beyond imaging alone. By integrating this diverse set of inputs, the system can offer more accurate diagnoses, especially when traditional imaging alone may not provide sufficient information to make a definitive diagnosis.

5. **Collaborate with Medical Experts**

Continuous collaboration with cardiologists, pediatricians, and radiologists is essential to refine the model’s clinical accuracy. Medical professionals bring invaluable domain knowledge and expertise that guide the feature selection process, model validation, and clinical interpretation of results. This collaboration is also crucial for improving the model’s robustness by incorporating medical context into its decision-making. For example, certain heart defects may not be immediately visible in imaging data but could be inferred from subtle changes in patient history or clinical symptoms. A medical expert can help refine the model by pointing out these nuanced connections, improving the overall prediction accuracy.

7.4 Scope of Future Work

1. **Expansion of Dataset Diversity**: Aggregate diverse echocardiogram, MRI, and CT datasets from multiple age groups (especially pediatric), geographic regions, and ethnicities. This ensures the model captures a wide variety of CHD presentations and anatomical variations, boosting generalizability.
2. **Integration of Multi-modal Data:**
   Enhance diagnostic accuracy by integrating diverse data sources, including echocardiography videos, cardiac MRI scans, electrocardiogram (ECG) signals, electronic health records (EHRs), and genetic profiles. Special emphasis should be placed on the temporal modeling of cardiac cycles using recurrent or transformer-based architectures to capture dynamic physiological patterns, enabling early detection of congenital anomalies and functional impairments.
3. **Deployment as a Web/Mobile Application:**
   Develop a secure, HIPAA-compliant web and mobile application tailored for pediatric cardiology and neonatal intensive care units (NICUs). The platform should enable real-time uploading and processing of cardiac images and videos, with embedded AI models providing instant diagnostic feedback. Key features should include offline capabilities for remote settings, automated report generation, integration with hospital information systems (HIS), and a clinician-friendly interface designed for high-stakes, time-sensitive environments.
4. **Model Optimization for Edge Devices**: Optimize models to run on portable or handheld ultrasound machines often used in rural and neonatal care. Lightweight models using techniques like pruning, quantization, and knowledge distillation can ensure high performance even on limited hardware.
5. **Explainable AI (XAI) Integration**.: Incorporate interpretable deep learning techniques, such as attention heatmaps on echocardiograms or saliency maps for MRIs. These tools can help cardiologists understand the model's focus areas, improving diagnostic confidence and patient-specific insights
6. **Continuous Model Training with Feedback**: Establish a secure feedback mechanism where pediatric cardiologists can review, validate, and annotate AI predictions. These corrections can be periodically used to retrain and refine the model, ensuring adaptability to evolving clinical standards and patient data.

APPENDIX

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt import os
import time

import seaborn as sns import
warnings
from sklearn.utils import resample
warnings.filterwarnings(action='ignore')
df=pd.read_csv(r"C:\Users\datma003\Desktop\projects1\Congenital Heart Disease
Detection\Dataset\critically ill pediatric patients in PICU.csv")
df

df['No. Patients'].nunique df.drop(columns=['No.
Patients'],inplace=True) df
df.isna().sum()
df.dropna(inplace=True) df
df.duplicated().sum()
df.drop_duplicates(inplace=True) df
df.describe()
df.info() cat_cols=[]
for i in df.columns:

    if df[i].dtype=='object':
        cat_cols.append(i)
cat_cols df['Age']
```

```
df['Age'] = df['Age'].str.extract('(\d+)').astype(int)
df df.info()
df['Age'].value_counts()
df['Genero'].value_counts() #Male -
1 and Female - 0
df['Genero']=df['Genero'].replace(['male','female'],[1,0])
df['Genero'].value_counts() df['Diagnosis'].value_counts()
df['Outcome'].value_counts()
# Dead- 1 and Survived -0
df['Outcome']=df['Outcome'].replace(['survived','dead'],[0,1])
df['Outcome'].value_counts()
df['Hour event (seconds)']
df['Hour event (seconds)'] = pd.to_datetime(df['Hour event (seconds)'], format="mixed", errors='coerce') if
df['Hour event (seconds)'].isnull().any():
   df['Hour event (seconds)'] = pd.to_datetime(df['Hour event (seconds)'].fillna(df['Hour event
(seconds)'].str.slice(0, 5)), errors='coerce')
# Convert times to minutes
df['Hour event in Minutes'] = df['Hour event (seconds)'].dt.hour * 60 + df['Hour event
(seconds)'].dt.minute
df['Hour event in Minutes']
df.drop(columns=['Hour event (seconds)'],inplace=True) cat_cols
df['oxygen saturation'].unique() #Normal
Oxygen Saturation level is 95
df['oxygen saturation']=df['oxygen saturation'].replace(['NORMAL'],[95]) df['oxygen
saturation']=df['oxygen saturation'].astype('int64')
df.info()
plt.figure(figsize=(15,5))
plt.subplot(1,3,1)
sns.boxplot(df['Age'])
plt.title("Age")
plt.subplot(1,3,2)
sns.boxplot(df['Weight (Kg)'])
plt.title("Weight") plt.subplot(1,3,3)
sns.boxplot(df['Height (cms)'])
plt.title("Height")
plt.figure(figsize=(15,5))
plt.subplot(1,3,1)
sns.boxplot(df['Heart Rate'])
plt.title("Heart Rate")


plt.subplot(1,3,2) sns.boxplot(df['oxygen
saturation']) plt.title("oxygen saturation")
plt.subplot(1,3,3)
sns.boxplot(df['Respiratory Rate'])
plt.title("Respiratory Rate")
plt.figure(figsize=(15,5))
plt.subplot(1,4,1)
sns.boxplot(df['Systolic Blood Pressure'])
plt.title("Systolic Blood Pressure") plt.subplot(1,4,2)
sns.boxplot(df['Diastolic Blood Pressure, '])
plt.title("Diastolic Blood Pressure") plt.subplot(1,4,3)
sns.boxplot(df['Mean Blood Pressure'])
plt.title("Mean Blood Pressure") plt.subplot(1,4,4)
sns.boxplot(df['Hour event in Minutes'])
plt.title("Hour event in Minutes")
outliers=['Age','Weight (Kg)','Height (cms)','Systolic Blood Pressure','Diastolic Blood Pressure, ','Mean
Blood Pressure']
```

```
other_cols=['oxygen saturation','Respiratory Rate'] for i
in other_cols:
    max=df[i].max()
    print(len(df))
    df=df[df[i]!=max]
    print(len(df))
for i in outliers:
    q1=df[i].quantile(0.25)
    q3=df[i].quantile(0.75)
    iqr=q3-q1
    print("For column ",i," Q1 is ",q1," and Q3 is ",q3)
    print(len(df)-len(df[(df[i]>=q1-(1.5*iqr))&(df[i]<=q3+(1.5*iqr))]),f" is the number of outliers that is being
removed from {i} column")
    df=df[(df[i]>=q1-(1.5*iqr))&(df[i]<=q3+(1.5*iqr))]
len(df)
df['Outcome'].value_counts() df_majority =
df[df['Outcome'] == 0] df_minority =
df[df['Outcome'] == 1]
df_majority_undersampled = resample(df_majority,replace=False,n_samples=len(df_minority)) balanced_df
= pd.concat([df_majority_undersampled, df_minority])


from sklearn.preprocessing import LabelEncoder from
sklearn.preprocessing import StandardScaler
le=LabelEncoder()
sc=StandardScaler()
df['Diagnosis']=le.fit_transform(df['Diagnosis']) df
balanced_df['Diagnosis']=le.fit_transform(balanced_df['Diagnosis'])



import joblib
joblib.dump(le,"le.pkl")
```

```
plt.figure(figsize=(15, 12))
for i, col in enumerate(['Heart Rate', 'oxygen saturation', 'Respiratory Rate',
                         'Systolic Blood Pressure', 'Diastolic Blood Pressure, ', 'Mean Blood Pressure'], 1):
    plt.subplot(3, 2, i)
    sns.scatterplot(data=df, x='Age', y=col, hue='Outcome')
    plt.title(f'Age vs {col}')
  plt.tight_layout()
  plt.show()
  time_cols = ['Heart Rate', 'Systolic Blood Pressure', 'Diastolic Blood Pressure, ', 'Mean Blood Pressure']
  plt.figure(figsize=(10, 6))
  for col in time_cols:
    sns.lineplot(data=df, x='Hour event in Minutes', y=col, label=col) plt.title('Vital
  Signs Over Time')
  plt.xlabel('Hour event in Minutes')
  plt.ylabel('Vital Sign Value')
  plt.legend()
  plt.show()
  plt.figure(figsize=(10, 5))
  for i, col in enumerate(['Genero', 'Diagnosis'], 1):
    plt.subplot(1, 2, i)
    sns.countplot(data=df, x=col)
    plt.title(f'{col} Distribution')
  plt.tight_layout()
  plt.show()
  scaling_cols=list(df.columns)
  scaling_cols
  scaling_cols.remove("Outcome")
  scaling_cols
  df[scaling_cols]=sc.fit_transform(df[scaling_cols]) df
balanced_df testing=pd.DataFrame(balanced_df.iloc[0])
```

```
testing.to_csv("testing.csv",index=False)


balanced_df[scaling_cols]=sc.fit_transform(balanced_df[scaling_cols])
joblib.dump(sc,"sc.pkl")
from sklearn.feature_selection import SelectKBest, f_classif, chi2 from
sklearn.feature_selection import chi2
def FS_KBest(df):
    col_list = []
    for col in df.columns:
        if ((df[col].dtype != 'object') & (col != 'Outcome')):
            col_list.append(col)
    xx = df[col_list]  yy =
    df['Outcome']
    fs = SelectKBest(score_func=f_classif, k=7) X_selected
    = fs.fit_transform(xx, yy)
    l = list(xx.columns)
    for i in range(0, len(l)): print(f"{i} =
        {l[i]}")
    return fs.get_support(True)
FS_KBest(df)
from sklearn.model_selection import train_test_split from
sklearn.linear_model import LogisticRegression from
sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier from
sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier from
sklearn.naive_bayes import GaussianNB
from sklearn.metrics import *
X=balanced_df.drop(columns=['Outcome']) Y=balanced_df['Outcome']
balanced_df x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,stratify=Y)
len(x_train),len(y_test)
```

```
lr1=LogisticRegression()

start_time=time.time()
lr1.fit(x_train,y_train)
end_time=time.time()
lrt=end_time-start_time
lr1_pred=lr1.predict(x_test)
print("Accuracy of Logistic Regression:",accuracy_score(lr1_pred,y_test))
print(classification_report(lr1_pred,y_test))
dt1=DecisionTreeClassifier()
start_time=time.time()
dt1.fit(x_train,y_train)
end_time=time.time()
dtt=end_time-start_time
dt1_pred=dt1.predict(x_test)
print("Accuracy of Decision Tree:",accuracy_score(dt1_pred,y_test))
print(classification_report(dt1_pred,y_test)) rf1=RandomForestClassifier()
start_time=time.time()
rf1.fit(x_train,y_train)
end_time=time.time()
rft=end_time-start_time
rf1_pred=rf1.predict(x_test)
print("Accuracy of Random Forest:",accuracy_score(rf1_pred,y_test))
print(classification_report(rf1_pred,y_test)) KNN1=KNeighborsClassifier()
start_time=time.time()
KNN1.fit(x_train,y_train)
end_time=time.time() KNNt=end_time-
start_time
KNN1_pred=KNN1.predict(x_test)
print("Accuracy of KNN:",accuracy_score(KNN1_pred,y_test))
print(classification_report(KNN1_pred,y_test)) GNB1=GaussianNB(
from flask import Flask, render_template, request, redirect, url_for, send_file
import pandas as pd
import numpy as np
import joblib
from sklearn.preprocessing import LabelEncoder, StandardScaler
import os

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'uploads'
os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)

# Load pre-trained model and encoders
MODEL_PATH = "dt1.pkl"
SCALER_PATH = "sc.pkl"
attack_mapping = {
    0: "Survived",
    1: "Dead"
}
model = joblib.load(MODEL_PATH)
scaler = joblib.load(SCALER_PATH)
scaling_cols=['Age',
 'Weight (Kg)',
 'Height (cms)',
 'Genero',
 'Diagnosis',
 'Heart Rate',
 'oxygen saturation',
 'Respiratory Rate',
 'Systolic Blood Pressure',
 'Diastolic Blood Pressure, ',
 'Mean Blood Pressure',
 'Hour event in Minutes']

# Preprocessing functions
def preprocess_data(df, scaler):
    # Drop unnecessary columns if present
    #testing=pd.read_csv("testing.csv")
    scaling_cols=['Age',
 'Weight (Kg)',
 'Height (cms)',
 'Genero',
 'Diagnosis',
 'Heart Rate',
 'oxygen saturation',
 'Respiratory Rate',
 'Systolic Blood Pressure',
 'Diastolic Blood Pressure, ',
 'Mean Blood Pressure',
 'Hour event in Minutes']

    scaled_values = scaler.transform(df[scaling_cols])
    scaled_df = pd.DataFrame(scaled_values, columns=scaling_cols)
    df[scaling_cols] = scaled_df
    df.drop(columns='Outcome',inplace=True)

    return df
```

Login page

```
from flask import Flask, render_template, request, redirect, url_for, send_file
import pandas as pd
import numpy as np
import joblib
from sklearn.preprocessing import LabelEncoder, StandardScaler
import os

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'uploads'
os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)

# Load pre-trained model and encoders
MODEL_PATH = "dt1.pkl"
SCALER_PATH = "sc.pkl"
attack_mapping = {
    0: "Survived",
    1: "Dead"
}
model = joblib.load(MODEL_PATH)
scaler = joblib.load(SCALER_PATH)
scaling_cols=['Age',
 'Weight (Kg)',
 'Height (cms)',
 'Genero',
 'Diagnosis',
 'Heart Rate',
 'oxygen saturation',
 'Respiratory Rate',
 'Systolic Blood Pressure',
 'Diastolic Blood Pressure, ',
 'Mean Blood Pressure',
 'Hour event in Minutes']

# Preprocessing functions
def preprocess_data(df, scaler):
    # Drop unnecessary columns if present
    #testing=pd.read_csv("testing.csv")
    scaling_cols=['Age',
 'Weight (Kg)',
 'Height (cms)',
 'Genero',
 'Diagnosis',
 'Heart Rate',
 'oxygen saturation',
 'Respiratory Rate',
 'Systolic Blood Pressure',
 'Diastolic Blood Pressure, ',
 'Mean Blood Pressure',
 'Hour event in Minutes']

    scaled_values = scaler.transform(df[scaling_cols])
    scaled_df = pd.DataFrame(scaled_values, columns=scaling_cols)
    df[scaling_cols] = scaled_df
    df.drop(columns='Outcome',inplace=True)

    return df
  #Login Page

@app.route('/', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        file = request.files['file']
        if file:
            file_path = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
            file.save(file_path)
            return redirect(url_for('predict', filename=file.filename))
    return render_template('index.html')

@app.route('/predict/<filename>')
def predict(filename):
    file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    df = pd.read_csv(file_path) if filename.endswith('.csv') else pd.read_excel(file_path)
    df_processed = preprocess_data(df, scaler)

    predictions = model.predict(df_processed)
    #df['Prediction'] = predictions
    df['Prediction'] = [attack_mapping.get(pred, "Unknown") for pred in predictions]
    results_path = os.path.join(app.config['UPLOAD_FOLDER'], 'results.csv')
    df.to_csv(results_path, index=False)

@app.route('/uploads/<filename>')
def download_file(filename)
if __name__ == '__main__':
    app.run(debug=True
```

REFERENCES

[1] L. Zhao et al., “Birth prevalence of congenital heart disease in China, 1980–2019: A systematic review and meta-analysis of 617 studies,” *Eur. J. Epidemiol.*, vol. 35, no. 7, pp. 631–642, Jul. 2020, doi: 10.1007/s10654-020-00653-0.

[2] C. I. Tchervenkov et al., “The nomenclature, definition and classification of hypoplastic left heart syndrome,” *Cardiol. Young*, vol. 16, no. 4, pp. 339–368, 2006, doi: 10.1017/S1047951106000291.

[3] B. R. White, D. J. Goldberg, and J. Rychik, “Hypoplastic left heart syndrome,” in *Echocardiography in Pediatric and Congenital Heart Disease: From Fetus to Adult*. Wiley, 2021, pp. 405–429, doi: 10.1002/9781119612858.ch21.

[4] J. Rychik, “Hypoplastic left heart syndrome: Can we change the rules of the game?” *Circulation*, vol. 130, no. 8, pp. 629–631, Aug. 2014, doi: 10.1161/CIRCULATIONAHA.114.011728.

[5] C. Ferencz et al., “Congenital heart disease: Prevalence at livebirth: The Baltimore-Washington infant study,” *Am. J. Epidemiol.*, vol. 121, no. 1, pp. 31–36, 1985, doi: 10.1093/oxfordjournals.aje.a113979.

[6] C. D. Morris, J. Outcalt, and V. D. Menashe, “Hypoplastic left heart syndrome: Natural history in a geographically defined population,” *Pediatrics*, vol. 85, no. 6, pp. 977–983, Jun. 1990, doi: 10.1542/peds.85.6.977.

[7] M. E. Abbott, *Atlas of Congenital Cardiac Disease: New Edition*. McGill-Queen’s Press, 2006. [Online]. Available: Google Books

[8] D. van der Linde et al., “Birth prevalence of congenital heart disease worldwide,” *J. Am. Coll. Cardiol.*, vol. 58, no. 21, pp. 2241–2247, Nov. 2011, doi: 10.1016/j.jacc.2011.08.025.

[9] R. B. Hinton et al., “Hypoplastic left heart syndrome is heritable,” *J. Am. Coll. Cardiol.*, vol. 50, no. 16, pp. 1590–1595, Oct. 2007, doi: 10.1016/j.jacc.2007.07.021.

[10] X. Liu et al., “The complex genetics of hypoplastic left heart syndrome,” *Nature Genet.*, vol. 49, no. 7, pp. 1152–1159, Jul. 2017, doi: 10.1038/ng.3870.