EARLY DETECTION OF HEART FAILURE AND EFFECTIVE OPTIMIZED FEATURE SELECTION USING ARTIFICIAL NEURAL NETWORKS.

Project Report

**Submitted in Partial Fulfillment of the Requirements for the Award of Degree of Bachelor of Technology
In**

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

By

D. ABHI PRIYA	21K61A6114
M. SESHUNAGU	22K65A6103
B. KEERTHAN KUMAR	21K61A6104
S. BALA SAI ASHOK KUMAR	21K61A6155

Under the esteemed guidance of

**Dr. Shaik Mohammad Rafee
Professor & Hod , AI&ML**

sasi autonomous INSTITUTE OF TECHNOLOGY & ENGINEERING TADEPALLIGUDEM

Accredited by NAAC with "A+" Grade
Recognised by UGC under section 2(f) &12(B)
Approved by AICTE - New Delhi
Permanently Affiliated to JNTUK, SBTET
Ranked as "A" Grade by Govt. of A.P.

Department of Artificial Intelligence and Machine Learning

SASI INSTITUTE OF TECHNOLOGY&ENGINEERING (Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada, and SBTET-Hyderabad, accredited by NAAC with 'A' Grade and NBA, ranked as "A" Grade by Govt. of A.P., Recognised by UGC 2(f) & 12(B)) Kadakatla, TADEPALLIGUDEM– 534 101

ACADEMIC YEAR 2023-2024

VISION AND MISSION OF INSTITUTE

VISION

Confect as a premier institute for professional education by creating technocrats who can address the society's needs through inventions and innovations.

MISSION

1. Partake in the national growth of technological, industrial, and industrial areas with societal responsibilities.
2. Provide an environment that promotes productive research.
3. Meet stakeholder's expectations through continued and sustained quality improvements.

VISION AND MISSION OF DEPARTMENT

VISION

To create competent Engineers with solid foundation in the domain of Artificial Intelligence and Machine Learning to contribute in uplifting of rural communities.

MISSION

1. To create an ambience that facilitates blended learning and extensive use of emerging technologies as a path forward to meaningful employment in the field of artificial intelligence and machine learning.
2. To facilitate collaborative learning with multi- displinary teams that encourage research intiatives leading to innovations.
3. To include among students the highest level of professional conduct and ethical values.

PROGRAM OUTCOMES (Pos)

Engineering Graduates will be able to :

PO1 : Engineering Knowledge:

Apply knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2 : Problem Analysis:

Identity, formulates, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3 : Design/ Development of Solutions:

Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

PO4 : Conduct investigations of complex problems:

Using research-based knowledge and research methods including design of experiments, analysis, and interpretation of data and synthesis of the information to provide valid conclusions.

PO5 : Modern Tool Usage:

Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6 : The Engineer and Society:

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7 : Environment and Sustainability:

Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

PO8 : Ethics:

Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

PO9 : Individual and Team Work:

Function effectively as an individual, and as a member or leader in diverse teams and multidisciplinary settings.

PO10 : Communication:

Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

PO11 : Project Management and Finance:

Demonstrate knowledge and understanding of engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 : Life-long Learning:

Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1. Students will be able to utilize core principles of Artificial Intelligence Engineering for the design, development and prototyping of AI Subsystems.

PSO2. Students will be able to employ acquired knowledge in data storage, data analytics, and Machine Intelligence to address and solve practical business challenges.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1 : Graduates will be able to apply the domain knowledge and the technological skills to gain meaningful employment and adapt to the ever demand of technological landscape.

PEO2 : Graduates will be able to pursue and excel in higher education and research.

PEO3 : :Graduates will be able to evolve as leaders exhibiting highest level of ethics.

COURSE OUTCOMES (COs)

CO1. Develop problem formation and design skills for engineering and real-world problems.

CO2. Collect and Generate ideas through literature surveys on current research areas which help to analyze and present to impart knowledge in different fields.

CO3. Impart knowledge of software & hardware to meet industry perspective needs and standards.

CO4. Create interest to research innovative ideas as lifelong learning.

CO5. Ability to work with a team, and enrich presentation and communication skills.

CO6. Create a platform that makes students employable.

EXPECTED OUTCOMES

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge
PO2: Problem Analysis
PO3: Design/Development of Solutions
PO4: Conduct an investigation of complex problems
PO5: Modern Tool Usage
PO6: The Engineer and Society
PO7: Environment and Sustainability
PO8: Ethics
PO9: Individual Teamwork
PO10: Communication
PO11: Life-long Learning
PO12: Project Management and Finance

SASI INSTITUTE OF TECHNOLOGY & ENGINEERING

(Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada, and SBTET-Hyderabad, accredited by NAAC with 'A' Grade and NBA, ranked as "A" Grade by Govt. of A.P., Recognised by UGC 2(f) & 12(B))

Kadakatla, TADEPALLIGUDEM– 534 101

Department of Artificial Intelligence and Machine Learning

...Values We Live...
sasi autonomous INSTITUTE OF TECHNOLOGY & ENGINEERING TADEPALLIGUDEM

Accredited by NAAC with "A+" Grade
Recognised by UGC under section 2(f) &12(B)
Approved by AICTE - New Delhi
Permanently Affiliated to JNTUK, SBTET
Ranked as "A" Grade by Govt. of A.P.

CERTIFICATE

*This is to certify that the project work entitled "**Early detection of heart failure and effective optimized feature selection using Artificial neural networks.**" is being submitted by D. ABHI PRIYA(21K61A6114), M. SESHUNAGU(22K65A6103), B KEERTHAN KUMAR (21K61A6104), S. BALA SAI ASHOK KUMAR (21K61A6155) in partial fulfillment for the award of the degree of **BACHELOR OF TECHNOLOGY**, in **Artificial intelligence and machine leaning** to Jawaharlal Nehru Technological University, Kakinada during the academic year 2024 to 2025 is a record of Bonafide work carried out by them under my/our found to be satisfactory. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any other degree or diploma.*

Project Supervisor	**Head of the Department**
Dr. Shaik mohammad Rafee	Dr. Shaik mohammad Rafee
Professor & HOD	Professor & HOD
Department of AI&ML	Department of AI&ML

External Examiner

DECLARATION OF THE CANDIDATES

We **D.Abhi Priya, (21K61A6114), M. Seshunagu, (22K65A6103), B Keerthan Kumar , (21K61A6104), S. Bala Sai Ashok Kumar, (21K61A6155),** here by declare that the project report entitled "**Early Detection of Heart Failure and Effective Optimized Feature Selection using Artificial Neural Networks.**" carried out under esteemed supervision of **Dr. Shaik Mohammad Rafee**, is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Artificial Intelligence and machine learning. This is are cord of work carried out by us and the results embodied in this project has not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

Project Associates

D. Abhi Priya	(21K61A6114)
M. Seshunagu	(22K65A6103)
B. keerthan kumar	(21K61A6104)
S. Bala Sai Ashok Kumar	(21K61A6155)

ACKNOWLEDGEMENT

We pay obeisance to our dynamic Chairman, **Sri B. Venu Gopala Krishna**, Sasi Educational Society, for his inspiring presence, which has always been the principal driving force behind all-over endeavors. First of all, we would like to extend a special thanks to **Sri. M. Narendra Krishna**, Vice-Chairman, Sasi Institute of Technology and Engineering, for his everlasting support.

It is true pleasure for us to thank **Dr. Mohamad Ismeal**, Principal, Sasi Institute of Technology and Engineering, who is the striving force for us to make this project through periodical audits.

We feel the privilege to thank **Dr. Shaik Mohammad Rafee**, Professor and Head of the Department, of Artificial Intelligence and Machine Learning, for providing us with invaluable feedback on our work, which allowed us to constantly improve our project.

We are very much grateful to thank our supervisor **Dr. Shaik Mohammad Rafee**, Professor and Head of the Department, Sasi Institute of Technology and Engineering for his constant encouragement, monitoring, and guidance and live with us throughout the submission of the project. He motivated us whenever we encountered an obstacle along the way.

We are very happy to thank our project coordinator **P. Sheela,** and all the Project Evaluation Committee members who provide valuable suggestions to finetune our ideas and projects. We are also thankful for all teaching and non-teaching staff members who contributed well to the successful completion of our project work.

With gratitude,

D. Abhi Priya (21K61A6114)

M. Seshunagu (22K65A6103)

B. keerthan kumar (21K61A6104)

S.Bala Sai Ashok kumar (21K61A6155)

ABSTRACT

Heart failure (HF) is a pervasive and life-threatening condition that affects millions of individuals worldwide, contributing significantly to global morbidity and mortality rates. As a chronic condition where the heart fails to pump blood effectively, it disrupts the supply of oxygen and nutrients to vital organs, leading to severe health complications. The growing prevalence of HF, driven by factors such as aging populations, lifestyle changes, and increasing rates of cardiovascular risk factors like hypertension and diabetes, underscores the urgent need for advanced diagnostic tools.

Traditional diagnostic methods, such as manual interpretation of clinical data or reliance on basic statistical models, often fall short in detecting HF at an early stage due to their inability to handle the complexity and nonlinearity of medical data. This research introduces a novel hybrid approach combining Artificial Neural Networks (ANNs) and Decision Trees (DT) to address these shortcomings, offering a robust framework for early detection and severity classification of heart failure. By leveraging the strengths of deep learning for feature optimization and decision trees for interpretable classification, this system aims to revolutionize HF prediction, enabling timely interventions that can save lives and reduce healthcare costs.

Keywords: Deep Learning, ANN, Machine Learning, Heart Failure

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

HF	-	Heart Failure
HA	-	Heart Attack

ANN	-	Artificial Neural Networks
DT	-	Decision Tree
SVM	-	Support Vector Machine
AI	-	Artificial Intelligence
UCI	-	University of California, Irvine.
CPT+	-	Compact Prediction Tree+
M.D	-	Doctor of Medicine
Ph.D	-	Doctor of Philosophy
RMSProp	-	Root Mean Square Propagation
ReLU	-	Rectified Linear Unit
UML	-	Unified Modeling Language
DFD	-	Data Flow Diagrams
HTTP	-	HyperText Transfer Protocol
DB	-	Database
API	-	Application Programming Interface
CSV	-	Comma-Separated Values
Dr	-	Doctorate

CHAPTER 1: INTRODUCTION

1.1 PREAMBLE

The heart serves as a essential part within a human body tasked with circulating the blood through the entire circulatory system. It functions as a muscular pump that ensures oxygen-rich blood reaches all the organs and tissues while also removing carbon dioxide and waste products. Positioned slightly to the left of the chest, the heart serves a central role in sustaining human life. Heart failure(HF) is a condition where the heart cannot effectively circulate the blood leading to heart problems.

Early diagnosis and Forecasting plays a vital position in enhancing patient outcomes and improve survival rates. Traditional methods for heart failure prediction often face challenges in handling complex medical data and capturing nonlinear relationships. However deep learning techniques have emerged as promising alternatives, offering enhanced predictive capabilities. Whereas Heart Failure (HF) should not be confused with a Heart Attack .

HA happens when a blood clot obstructs one of the heart's vessels, impairing the heart's ability to pump blood effectively. In such situations, the heart may struggle to supply oxygenated blood to tissues, resulting in oxygen deprivation throughout the body. The severity of this circulatory failure typically depends on the extent of the heart attack. A heart attack (HA) is among the leading causes of HF. Key contributors to HF include coronary artery disease, inflammation of the heart, hypertension (high blood pressure), cardiomyopathy, or irregular heart rhythms. Additionally, other contributors such as sudden shocks, intense stress may precipitate either HF or heart attack. These events are inherently unpredictable. Deep learning is a sub part of AI which is used for finding patterns in large datasets. The artificial neural networks (ANNs) are the most important part in the deep learning concept and it works as like a human brain.

The ANNs consists of interconnected layers of neurons in the network each processing information and passing it on to subsequent layers. This hierarchical structure empowers ANNs to extract complex features including image recognition, natural language processing and in this case heat failure prediction. The main advantage of ANNs is their capability of capturing the non-linear association between the various factors providing a more accurate depiction of the underlying biological process involved in heart failure and also ANNs can handle large and various datasets, and obliging a wide range of clinical variables.

1.3 PURPOSE OF THE WORK

The whole point of this work is to build a smart tool that catches heart failure early, before it gets bad. We're mixing two helpers Artificial Neural Networks (ANNs) and Decision Trees (DTs) to look at patient info, like the Cleveland Heart Disease dataset. ANNs will dig through 14 things, like age or blood pressure, and pick the 8 most important ones. Then, DTs will sort out who's at risk of heart failure and who's not, nice and clear. This way, doctors get a quick, reliable way to spot trouble early. It's all about helping patients get care sooner and keeping healthcare costs down by avoiding big problems later.

1.4 SIGNIFICANCE OF THE STUDY

This study is significant as it addresses a major global health concern heart failure. Over 26 million people suffer from it worldwide, and early detection is key. The hybrid ANN-DT model brings together accuracy and interpretability. It empowers doctors with fast, reliable insights that help prevent complications. The system is designed to be easily integrated into hospitals and clinics. It reduces the need for expensive and late-stage interventions. Patients benefit from personalized, timely care. Healthcare systems benefit from reduced resource strain. It's also useful for training future doctors and tech experts. This study offers a scalable, tech-based health solution with wide impact.

1.4.1 Patient Safety and Well-being

Early prediction of heart failure significantly enhances patient safety. It reduces the chances of serious complications or emergency interventions. Patients can start simple treatments like medications or lifestyle changes. This reduces physical and emotional suffering. Doctors can make timely decisions based on data-driven insights. It keeps patients healthier and more active in daily life. Early care lowers hospitalization and increases life expectancy. It gives patients peace of mind through early awareness. The system supports preventive care over reactive treatment. This makes a direct, positive impact on patient well-being.

1.4.2 Addressing a Growing Issue

Heart failure is a rising concern due to aging populations and lifestyle diseases. With millions at risk, early detection is more important than ever. This project directly tackles the issue using AI tools. It reduces the burden on hospitals and caregivers. By finding risks early, treatment is easier and more effective. It helps manage healthcare costs and resource allocation. The model can assist in both urban and rural clinics. It scales easily to serve larger or diverse populations. It provides a proactive solution to a growing problem. That's essential in today's global healthcare landscape.

1.4.3 Technological Advancements

This system introduces a smart use of AI in healthcare prediction. ANNs offer deep pattern recognition from complex patient data. DTs provide clear and easy-to-understand output. Together, they form a powerful yet simple decision-making tool. This hybrid approach improves both accuracy and explainability. It demonstrates the practical use of AI in real-world medicine. The system can be a model for future AI-driven solutions. It helps bridge the gap between data science and clinical needs. It's a leap forward from traditional diagnostic tools. This project marks innovation in preventive digital healthcare.

1.4.4 Preventive Measures and Early Intervention

Prevention is more effective and affordable than late treatment. This system helps detect heart failure before symptoms worsen. Doctors can recommend lifestyle changes or low-risk medication early. It avoids the need for surgery or emergency hospitalization. Patients experience less stress and fewer health complications. It supports long-term wellness and reduces relapses. Medical staff can act quickly based on alerts. This makes the treatment process smoother and more effective. Early intervention is the foundation of preventive medicine. This project embodies that principle with intelligent support.

1.4.5 Building Trust in Healthcare Systems

Trust is key in adopting healthcare technologies. The Decision Tree component offers transparency in results. Doctors and patients can see the logic behind each prediction. This makes the system easier to trust and use. It replaces guesswork with reliable, explainable outcomes. The tool supports human decision-making, not replaces it. It shows how AI can work side by side with clinicians. That boosts confidence in tech-aided healthcare systems. The more transparent and accurate the tool, the better the trust. This trust drives better patient engagement and system adoption.

1.4.6 Educational Impact

This system serves as a valuable educational resource. Medical students learn how AI supports diagnostics. Engineering students see practical healthcare applications of machine learning. It demonstrates how tech and medicine can collaborate effectively. Institutions can use this as a real-world case study. It promotes interdisciplinary education and skill development. Students get exposure to real healthcare data. The project builds awareness of ethical AI in healthcare. It also sparks innovation for future AI tools. Learning is more impactful when tied to real applications.

1.4.7 Collaboration with Medical Platforms

Our tool can plug right into hospital computers, helping doctors on the spot. It gives real-time alerts while they're with patients, speeding things up. It works with records already there, so it's a team player in the clinic. Even small offices can use it, not just big places. This connection makes decisions quicker and care better. It's like a bridge between tech and medicine that really works.

1.4.8 Global Reach and Impact

This system starts with data like Cleveland's but can fit anywhere in the world. It doesn't care where you're from it can help all kinds of people. With millions at risk, it's got the power to make a huge difference. It's flexible enough to adapt to different countries and health setups. Doctors everywhere could use it to fight heart failure. That's a global win for keeping folks healthy.

1.4.9 Data-Driven Insights

The tool shows doctors what matters most like cholesterol or age for heart failure risks. They can use that to plan care that fits each patient just right. It’s not guessing it’s based on solid info from the data. This makes treatments smarter and more personal. Patients get help that’s tailored, not one-size-fits-all. It’s a big step toward better, data-smart medicine.

1.4.10 Empowering Healthcare Providers

Doctors get a fast, trusty sidekick with this system, making their job easier and sharper. They don’t have to wade through messy data it’s all clear and ready to go. This lets them focus on helping patients, not puzzling over numbers. It’s like a superpower for spotting heart failure early. They can save more lives with less stress. That’s real power for healthcare pros.

1.5 BACKGROUND STUDY

To get why this project’s a good idea, we need to know where heart failure comes from and what’s been tried before. It’s a big problem that hits people hard and costs a lot, so finding it early is key. Old ways to predict it aren’t great, but new tech like deep learning is shaking things up. That’s why we’re mixing ANNs and DTs it’s a fresh take on an old fight. Here’s the full backstory to set the stage.

1.5.1 Emergence of Heart Failure

Heart failure shows up when the heart gets weak from stuff like clogged arteries or high blood pressure. It can’t pump blood right, so organs miss out on oxygen and nutrients. Things like heart muscle damage or past heart attacks make it worse over time. It sneaks up from lifestyle choices or other health issues too. People end up struggling because the heart just can’t keep up. That’s where it all starts, and why we need to catch it.

1.5.2 Impact on Individuals and Society

For folks, heart failure means feeling wiped out, short of breath, and not living as long. It’s tough to do normal stuff like walking or working, which hits hard emotionally too. For society, it’s a money drain hospitals spend tons on treatments and care. Families feel the stress of helping loved ones through it. With millions affected, it’s a load on everyone everywhere. That’s why fixing it matters so much.

1.5.3 Current Approaches to HF Prediction

Right now, doctors use basic math tools like stats or simple computer models like SVM and KNN to guess heart failure. These look at patient info but often miss tricky patterns in the data. They’re okay for small, clean sets, but real-life stuff is messier. Accuracy tops out around 76% to 90%, which isn’t good enough. Old ways just don’t cut it for something this serious. We need something better to step up.

1.5.4 Rise of Deep Learning in Healthcare

Deep learning, like ANNs, is a game-changer because it can handle big, jumbled medical records. It finds hidden clues that old tools skip, making it great for spotting diseases early.

Hospitals are starting to use it for stuff like cancer or heart problems. It’s fast and smart, digging through data humans can’t sort alone. That’s why it’s perfect for our heart failure fight. It’s the new wave we’re riding.

1.5.5 Motivation for Hybrid Framework

We’re mixing ANNs and DTs because old methods leave gaps like missing patterns or being hard to explain. ANNs are awesome at finding what matters in data, but they’re tricky to understand. DTs make it simple with clear yes/no answers doctors can trust. Together, they’re stronger smart and easy to use. That’s our push: better predictions that make sense. It’s why we’re doing this hybrid thing.

1.6 EVALUATION OF EXISTING SOLUTIONS

Let’s see what’s out there already for predicting heart failure and why it’s not enough. Old tools have some good points, but they stumble in big ways too. They can’t keep up with messy data or give doctors what they need fast. Here’s a close look at where they fall short and why we need a new plan.

1.6.1 Data Bias and Generalization Issues

Most models use tiny datasets like Cleveland with just 297-303 people, which skews things. They work okay there but flop with bigger, different groups. That bias means they don’t fit the real world well. It’s like training on one town and expecting it to work everywhere. We need something that stretches further. Small data just won’t cut it anymore.

1.6.2 Computational Complexity

Fancy setups like ensemble models or CPT+ need tons of computer juice to run, which slows them down. In a busy hospital, that’s a problem doctors can’t wait around. They’re heavy and clunky, not practical for quick checks. Simple tools are faster but miss too much. We need a balance that’s speedy and smart. Old ways are too much or too little.

1.6.3 Limited Feature Optimization

Older methods don’t trim out junk data well like extra stuff that muddies the picture. They keep noisy bits that mess up predictions, making them less sharp. Good feature picking is key, but they struggle with it. That leads to mistakes we can’t afford. Our tool needs to focus on what counts. Current ones just don’t clean up right.

1.6.4 Accuracy Constraints

The best these models hit is 76% to 90.5% accuracy, which sounds okay but isn’t for heart failure. Missing even a few cases can be a big deal when lives are on the line. They’re stuck in that range, leaving room to do better. Doctors need near-perfect to trust it fully. Old tools aren’t cutting close enough. We’ve got to aim higher.

1.7 UNADDRESSED CHALLENGES

Some big hurdles still stand in the way of perfect heart failure prediction. Old systems can't grow to handle huge datasets from real hospitals they're stuck small. Messy, real-world info trips them up, and they're not clear enough for doctors to use easily. Plus, making them super accurate without slowing down or confusing anyone is tough. We're aiming to fix these gaps with our new idea. It's a challenge worth taking on.

1.8 AIM & OBJECTIVE OF THE PROPOSED METHOD

1.8.1 Aim

Our big aim is to make a top-notch tool that catches heart failure early with a smart mix of tech. We want it to be super accurate and help doctors save lives fast. It's about using ANNs and DTs together to spot risks before they blow up. That's the dream driving us forward. Lives depend on getting this right.

1.8.2 Objective

We've got clear goals to make this work happen right. First, use ANNs to pull out the best features from patient data so we're not lost in extras. Second, let DTs sort folks into "heart failure" or "no heart failure" with top accuracy. Third, beat other models at how well and fast it predicts trouble. It's about making a tool doctors love to use. That's our mission in a nutshell.

1.9 MOTIVATION & SCOPE

1.9.1 Motivation

We're fired up to do this because catching heart failure early can save so many lives. Seeing millions struggle with it pushes us to help out. Cutting down big medical bills is a bonus that keeps us going. It's a chance to make a real mark on a tough problem. That hope and impact drive every step we take.

1.9.2 Scope

This system starts with clean data like Cleveland's, but it's built to grow bigger someday. It's focused on heart failure now, with an eye on real-time hospital use later. We're keeping it simple but ready to stretch to other datasets. It's a solid start with room to expand. That's where we're aiming for now.

1.10 APPLICATIONS

This tool can shine in lots of places like helping doctors spot heart failure fast in clinics. It can plug into hospital systems for instant alerts during checkups. It's great for planning personal care or teaching students about smart tech. Even small towns with basic setups could use it to boost health. Anywhere hearts need watching, it's got a job to do.

1.11 INITIAL ASSUMPTIONS

We’re betting the Cleveland data is solid enough to kick things off right. We think ANNs and DTs will play nice together and give good results. Doctors will jump on board if it’s accurate and simple, we hope. The data won’t be too messy to handle at first. If these don’t hold, we’ll tweak as we go. It’s our starting line.

1.12 TECHNICAL TERMS DEFINITIONS

Here’s what some big words mean simply: ANN is a computer brain that learns patterns, like heart failure signs. DT is a rule tree splitting data into yes/no, easy to follow. Feature means a data bit, like blood pressure, we check. Accuracy is how often we guess right about heart failure. Optimization is picking the best stuff to focus on. That’s the tech talk made clear.

1.13 ORGANIZATION OF THE REPORT

This report kicks off with this intro, laying out the why and what of our work. Next, we dig into past studies, then list what we need to build it. We’ll explain how it’s made, show the design, and test it out step-by-step. It wraps up with results and what’s next all you need to get the full picture. It’s our roadmap from start to finish.

1.14 SUMMARY

This chapter sums up why heart failure’s a huge deal and how old ways to predict it aren’t good enough. We’re bringing a new ANN-DT tool to find it early, mixing smarts with clarity. It’s about helping doctors, saving lives, and cutting costs with a better system. The plan’s laid out, and the stakes are high. That’s the heart of what we’re doing here.

CHAPTER 2: LITERATURE SURVEY

2.1 PREAMBLE

This part sets the stage by looking at what's been done to predict heart failure and why our work matters. It digs into how heart failure ties to patient data, why spotting it early is a big deal, and what others have tried so far. We'll see where those old ways stumble and how new tech like deep learning is shaking things up. Then, we'll figure out what's missing and why our hybrid ANN-DT idea could fill that gap. It ends with what we're aiming to do with this study.

2.1.1 Introduction to Heart Failure & Clinical Data Analysis

Heart failure occurs when the heart is no longer able to pump blood efficiently to meet the body's needs, leading to symptoms like fatigue, breathlessness, and fluid buildup. This condition has become a major global health issue, currently affecting over 26 million people, with numbers expected to rise due to aging populations and lifestyle-related factors. To detect heart failure early, doctors rely on clinical data such as a patient's age, cholesterol levels, blood pressure, and other vital health indicators. One of the most commonly used datasets for such research is the Cleveland dataset, which contains key medical attributes used to analyze heart disease risks. By studying this data through intelligent algorithms, patterns can be uncovered that help identify patients who are at high risk even before symptoms become serious. This predictive approach shifts healthcare from reaction to prevention, which can ultimately save lives. Our research builds on this concept by using smarter tools like artificial neural networks (ANN) to make the predictions more accurate, and decision trees (DT) to make the results more understandable to doctors. The goal is to transform raw clinical data into clear, timely, and life-saving insights.

2.1.2 Importance of Heart Failure Prediction

Heart failure remains one of the leading causes of mortality and hospitalization worldwide, affecting over 26 million people. Early detection plays a critical role in preventing disease progression and reducing the need for invasive, expensive treatments such as surgeries or prolonged hospital stays. Accurate prediction models enable doctors to identify at-risk individuals before symptoms worsen, allowing for timely interventions like lifestyle changes, medications, or routine monitoring. This not only improves patient quality of life but also significantly cuts down healthcare costs associated with emergency care and chronic management. In many cases, early prediction can mean the difference between manageable conditions and life-threatening episodes. As aging populations and unhealthy lifestyles continue to fuel the rise in heart-related conditions, having reliable tools for early diagnosis becomes more urgent than ever. Timely action driven by predictive models ensures that care is proactive rather than reactive. Ultimately, heart failure prediction isn't just about data it's about saving lives, enhancing outcomes, and improving healthcare efficiency.

2.1.3 Existing Approaches of Heart Failure Prediction

Over the years, researchers have explored a variety of techniques to predict heart failure using computational models. Traditional methods like logistic regression and decision trees have been widely used due to their simplicity and interpretability. These models generally operate on datasets such as the Cleveland heart disease dataset, which contains around 297 to 303 records. While they provide a foundational understanding of risk factors, their predictive power is often limited due to assumptions like linearity or independence between features. Other machine learning methods like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) have also been applied, offering improved performance in some cases, but they come with their own set of drawbacks like high computation time or sensitivity to feature scaling. In an effort to improve results, some studies have adopted ensemble techniques, combining multiple algorithms to increase accuracy. These can perform better than single models but often become too complex, harder to interpret, and computationally expensive. More recently, researchers have turned toward deep learning models like Artificial Neural Networks (ANNs), which excel at detecting complex, non-linear relationships in the data. However, while ANNs can dig deeper into patterns, they typically lack explainability a major concern in medical applications where understanding the rationale behind predictions is critical. Additionally, many of these studies rely heavily on small datasets, limiting the generalizability of the models. Overall, while existing approaches have made strides in prediction, challenges related to speed, interpretability, scalability, and dataset limitations continue to hinder their real-world applicability, motivating the need for more balanced and hybrid solutions.

2.1.4 Challenges in Existing Approaches

Despite significant progress in heart failure prediction using machine learning and deep learning models, several challenges remain in existing approaches. Most studies suffer from limited dataset sizes, lack of interpretability, slow processing times, or unrealistic assumptions about the data. These limitations hinder the practical deployment of such models in real-world healthcare settings. Below, we explore the specific challenges faced by key studies in detail.

Challenges in Mahgoub's et al:

Mahgoub's work applied the CPT+ deep learning model on the Cleveland dataset and achieved a promising accuracy of 90.5%. However, this model is computationally expensive and slow, especially when dealing with large-scale datasets. The Cleveland dataset used here only has 207 rows, which limits the generalizability of the model in real-world healthcare environments where much larger and more diverse datasets are involved. CPT+ is complex in its architecture and often hard to interpret for medical professionals who require quick and transparent decisions. Additionally, the model doesn't perform feature reduction or selection, making it prone to overfitting and irrelevant data noise. Its black-box nature adds another limitation it gives high performance but fails to explain how and why a prediction is made. In practical healthcare settings, especially in high-pressure environments like emergency care, this lack of interpretability becomes a serious bottleneck. Furthermore, training CPT+ demands high computational power and GPU resources, which may not be available in standard medical setups. While it's powerful, it's not agile or scalable without significant infrastructure. This makes it more academic than practical in its current form.

Challenges in Shouman's et al:

Shouman employed a combination of K-Means clustering and Decision Trees on a 297-row Cleveland dataset. The method tried to enhance classification using unsupervised learning, but it faced several practical drawbacks. Clustering methods like K-Means are highly sensitive to the initial centroid values; poor initialization can skew the entire model. In medical data where class boundaries are fuzzy and complex, K-Means may misclassify critical patient cases. Also, it assumes clusters are spherical and equally sized, which is rarely true in health records. While the method used Decision Trees for interpretability, the preceding clustering step introduced instability. The model also did not incorporate any advanced feature selection technique, which often resulted in noisy data impacting the classifier's performance. Moreover, the dataset's limited size restricts the model's performance on larger or more diverse populations. This makes it less reliable when scaled up. Shouman's system lacks adaptability to real-world, continuous patient data streams and doesn't factor in data imbalance a key issue in medical prediction where positive cases are often fewer. The method also doesn't support real-time processing, which limits its use in high-speed clinical settings. Overall, the technique was insightful but not optimized for clinical deployment.

Challenges in Priyanga's et al:

Priyanga utilized the Naïve Bayes algorithm with a weighted average variation (NBwa) on the standard Cleveland dataset. Although the model was lightweight and fast, it relied heavily on the strong assumption that features are conditionally independent. In reality, medical parameters such as blood pressure and cholesterol are often correlated. Ignoring these relationships led to oversimplification of complex interactions among clinical attributes, reducing the model's reliability. The accuracy achieved was 72.91%, which falls short for clinical safety standards where precision is critical. The model's feature engineering lacked advanced selection techniques and instead applied basic weighting, which failed to prioritize the most influential predictors. Additionally, NBwa doesn't support non-linear relationships, which are common in heart-related medical data. The model did not handle missing values or outliers well, making it less robust for real-world electronic health records. Furthermore, its performance was not benchmarked against modern deep learning models, so its relative effectiveness remains unclear. It also didn't address data imbalance issues that could skew classification results. Priyanga's approach is fast but too basic for the complex nature of heart failure prediction, and not scalable for future extensions involving larger, dynamic datasets.

Challenges in Marbaniang's et al:

Marbaniang tested multiple machine learning models including KNN, SVM, Naïve Bayes, and Random Forest, using UCI heart disease datasets. While this comparative approach was thorough, it lacked consistency and optimization in feature selection and data preprocessing. The study relied heavily on raw data without applying any dimensionality reduction, which increased noise and computational complexity. The highest accuracy recorded was 85.49%, which is good but not excellent. The models used failed to generalize well across different test scenarios, especially because the feature engineering was minimal. Additionally, algorithms like KNN are sensitive to irrelevant features and scaling issues, which were not addressed in the paper. The lack of interpretability in SVM and Random Forest also poses a challenge for clinical use. Moreover, the models did not focus on minimizing false negatives, which is critical in heart failure where missing a case could be fatal. They also struggled with handling real-time or streaming data, limiting their practical deployment. While the study offered a wide view of model performances, it lacked depth in tuning and integrating models in a way that would make them clinically actionable. Overall, the approach was informative but too generic and lacked innovation in feature handling and model synergy.

Challenges in Agrawal's et al:

Agrawal employed ensemble learning by combining 10 classifiers including Logistic Regression, Naïve Bayes, and Random Forest, achieving a decent 85.2% accuracy. While ensemble methods can often boost accuracy, they also introduce complexity. Agrawal's model suffered from increased computational load, which makes it unsuitable for real-time clinical environments. The ensemble model also lacked interpretability due to the combination of diverse algorithms, making it hard for healthcare professionals to understand or validate individual predictions. The dataset used (270 records) was relatively small and did not represent a wide variety of patient profiles, limiting the model's generalizability. Furthermore, the system had no built-in mechanism for intelligent feature selection or dimensionality reduction, increasing the risk of overfitting. While it performed slightly better than individual models, the trade-off in speed and transparency was significant. The study did not explore how well the ensemble could scale with larger datasets or adapt to imbalanced data scenarios. Agrawal's method is robust in theory but heavy and opaque in practice. This makes it less feasible in situations where explainability and speed are just as important as accuracy. It demonstrates the strengths of ensemble methods but also reveals the urgent need for more streamlined and interpretable solutions.

2.1.5 Advancements in Deep Learning for Healthcare

Deep learning, like ANNs, is changing healthcare by handling big, jumbled data better than old tools. It finds hidden clues like how cholesterol links to heart risks that others miss. Hospitals use it for stuff like cancer or heart issues because it's fast and smart. It's perfect for messy medical records with lots of factors. Studies show it can push accuracy higher than basic models. That's why we're tapping into it to make heart failure prediction sharper and quicker.

2.1.6 Literature Gap and Research Motivation

Past work leaves gaps like not picking the best data points or explaining results clearly to doctors. Many stick to small datasets or slow methods that don't fit real-world needs. Accuracy isn't high enough, and some tools are too tricky to use fast. We're motivated to fix this with a mix of ANNs for smart digging and DTs for clear answers. No one's nailed this combo for heart failure yet, so that's our chance. It's about making a tool that's both powerful and practical.

2.1.7 Objectives of the Study

Our study wants to build a hybrid ANN-DT system that predicts heart failure early and well. We'll use ANNs to shrink 14 data points from Cleveland into 8 key ones that matter most. Then, DTs will sort out who's at risk with clear, trusty results. We aim to beat old models in accuracy pushing past 90% and speed. It's about giving doctors something they can use right away to help patients. That's the plan we're chasing with this work.

2.2 RELATED WORK

Lots of smart people have tried predicting heart failure, and we've looked at their ideas to see what works and what doesn't. They've used everything from basic math to fancy computer tricks, mostly on the Cleveland dataset. Some hit decent accuracy, but there's always a catch like slow speed or small data limits. Here's a rundown of six key studies we've checked out, showing where they shine and stumble. It helps us figure out how to do better with our ANN-DT mix.

- **C. B. Rjeily [2]**: This study used a CPT+ algorithm on the Cleveland dataset with 207 rows to predict heart failure. It got a solid 90.5% accuracy, which sounds good at first glance. But that tiny dataset doesn't cover everyone worldwide it's too small to trust everywhere. Plus, CPT+ takes a lot of computer power, so it'd bog down on bigger data. It's a strong start, but not practical for real hospitals needing fast answers. That's a big hint for us to keep things quick and broad.

- **Shouman M [4]** : Here, they worked with Cleveland's 297 records, using K-Means clustering and DTs for prediction. They hit 83.9% accuracy, which is okay but not amazing. Picking the starting points for clustering was tricky, and their methods sometimes mixed up cases big and small values got jumbled. That risks missing heart failure in some folks. The small data size limits it too, so it's not ready for the big leagues. We're aiming higher and clearer with our approach.

- **Priyanga [5]** : This one used Naïve Bayes on a 303-record Cleveland set, scoring 86% accuracy. It's simple and assumes all data points don't affect each other, which isn't true for stuff like blood pressure and cholesterol. That assumption makes it less reliable for real medical data. They didn't test it against fancier models, so we don't know how it stacks up. The small dataset holds it back too. We want something that handles connections better.

- **Abdalla Mahgoub [1]**: This study compared a few models, with Multilayer Perceptron (an ANN type) hitting 85.49% accuracy on heart disease data. They found the ReLU function worked best, beating out KNN and Naïve Bayes in their tests. It's a nod to ANNs being strong, which we like. But it didn't mix in DTs for clarity or test on huge datasets. It's a solid push, but we're taking it further with our hybrid twist. That combo could be the key.

- **Marbaniang [9]**: They tested six models for heart disease, with KNN topping out at 72.91% and Random Forest at 72.12%. SVM was slow and heavy, while Naïve Bayes lagged at 60.1% pretty low. Adding BMI and blood pressure bumped accuracy a bit, which is neat. But those numbers are way below what we need for heart failure. It shows basic models struggle, so we're betting on deeper tech to lift us up. Speed and power are must-haves.

- **H. Agrawal [6]**: This one mixed 10 classifiers into an ensemble, getting 85.2% accuracy and 87.5% recall on heart failure. Logistic Regression (86.4%) and Random Forest (86.2%) did well, but Naïve Bayes lagged at 82.5%. The ensemble helped, but it's complex and risks overfitting too tuned to its data. It takes a lot of computing juice too, slowing it down. We're inspired but want simpler, faster results with our ANN-DT plan. Clarity's our edge here.

2.4 GAP IDENTIFIED

Despite numerous advancements in heart disease prediction using machine learning and deep learning, there remain significant gaps that current models have not addressed effectively. Many of the existing models either fail to explain their decisions clearly, rely on small and limited datasets, or are too computationally heavy for real-time, practical use. Deep learning models are powerful in uncovering hidden patterns, but their "black-box" nature makes them difficult to interpret in clinical settings. Our proposed hybrid model combining Artificial Neural Networks (ANNs) with Decision Trees (DTs) aims to bridge these critical gaps by enhancing both performance and interpretability. It offers a novel step forward by leveraging the strengths of both approaches, making it suitable for real-world medical applications.

2.4.1 Gap in Existing Literature

A review of existing literature shows that most studies peak at around 90.5% accuracy, with many relying solely on the Cleveland dataset, which contains just 303 records. These limited datasets pose serious challenges for real-world generalization. Moreover, there is little evidence of smart feature selection techniques being used to improve model efficiency. Heavy models like CPT+ or ensemble methods are often accurate but too slow for real-time decision-making, and their predictions are difficult for clinicians to interpret quickly. Critically, no study has attempted to integrate ANNs for deep learning with DTs for clear decision-making. This leaves a major gap where performance, clarity, and speed are all needed simultaneously that our research seeks to fill.

2.4.2 Rationale for Identified Gap

Small datasets, such as those with 207 or 270 rows, do not offer the statistical robustness required for broader clinical deployment. Additionally, simple models like Naïve Bayes, though fast, fail to capture complex feature interactions. On the other hand, ensemble and deep models often trade off speed for accuracy, making them less practical in fast-paced environments like hospitals. More importantly, physicians often struggle to trust predictions they cannot interpret, which is a major limitation of current deep learning approaches. Our hybrid model seeks to overcome this barrier by combining the pattern-recognition ability of ANNs with the interpretability of DTs, making the predictions both powerful and understandable.

2.4.3 Need for Hybrid ANN-DT Approach

Artificial Neural Networks are highly effective in identifying subtle, non-linear relationships in medical data, such as those indicative of heart disease. However, they are often criticized for their lack of transparency. On the flip side, Decision Trees offer simple, rule-based outputs that are easy to follow but may not capture deeper patterns on their own. By combining the two, we aim to harness the deep learning capabilities of ANNs for feature selection and complex pattern recognition, and then use DTs to present the results in a clear, interpretable format. This hybrid model surpasses older approaches like K-Means, KNN, or simple clustering models, which fail to deliver both accuracy and usability. Thus, the ANN-DT combination is not just innovative it is necessary.

2.4.4 Research Questions to Address Gap

To close the identified gap, this study seeks to answer several critical questions:

- Can ANNs effectively identify the top 8 most relevant features from a set of 14, leading to improved performance?
- Will Decision Trees make the resulting predictions clear and interpretable for medical professionals?
- Can the proposed hybrid model maintain high accuracy and speed on smaller datasets like Cleveland’s, while also being scalable to larger ones?
- How does the ANN-DT model compare with existing solutions like CPT+ or ensemble methods in terms of both performance and usability?
- Can this model resolve the issues of slow processing and small data reliance that past studies left unresolved?

These research questions are central to validating the proposed approach and demonstrating its real-world applicability.

2.4 PROBLEM STATEMENT

Heart failure (HF) is a critical health condition affecting millions of individuals globally and is often diagnosed only at an advanced stage. Traditional methods for heart failure prediction, such as statistical tools and conventional machine learning models like Logistic Regression and Support Vector Machines (SVM), often fail to deliver sufficient accuracy. These models struggle to handle the complex, high-dimensional, and often noisy nature of medical data, which includes variables such as age, cholesterol levels, and blood pressure. Moreover, they lack robust feature selection mechanisms, leading to issues like overfitting or underfitting, and frequently produce results that are difficult for healthcare professionals to interpret. This project aims to address these challenges by developing an efficient, accurate, and interpretable system for early prediction of heart failure. The proposed solution leverages Artificial Neural Networks (ANN) for optimal feature extraction and Decision Trees (DT) for reliable and understandable classification, thereby improving early detection and potentially reducing mortality and healthcare costs.

2.5 PROPOSED METHOD

The proposed method introduces a hybrid system that integrates the strengths of Artificial Neural Networks (ANN) and Decision Tree (DT) algorithms for accurate heart failure prediction. The core idea is to process the Cleveland Heart Disease dataset, which consists of 14 clinical features, and extract the most influential attributes using ANN. The ANN model acts as a powerful feature selector, reducing the dimensionality of the input from 13 attributes (excluding the target) to 8 optimized features. These selected features capture the most critical patterns and relationships necessary for accurate classification.

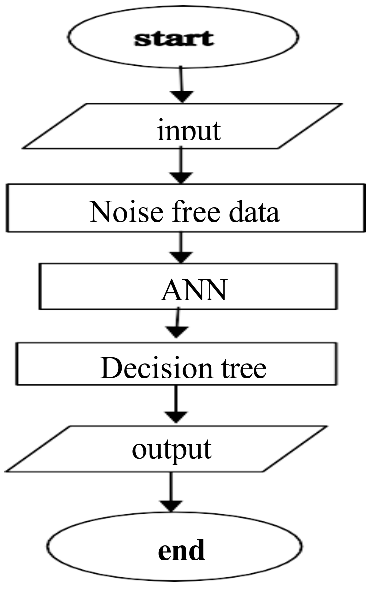


Figure 2.5.1 :- Detailed proposed Algorithm

This two-phase hybrid approach addresses the limitations of traditional single-model methods by combining deep learning's pattern recognition ability with the interpretability and simplicity of decision trees. The ANN handles complex, nonlinear dependencies in the data, while the DT offers transparent, rule-based classification. This results in a model that is not only highly accurate but also practical for real-world clinical use. The system is designed for both reliability and speed, allowing medical professionals to make quick and informed decisions. Ultimately, this method improves prediction accuracy while maintaining clarity and efficiency, making it a robust decision-support tool in the healthcare domain.

The proposed algorithms follow these steps to identify the heart failure outcome effectively. Initially the input taken from UCI repository then preprocess it to avoid noise data. The preprocessed data passed to Ann model, it process the data and extract the optimized features In other words reduce the input data 13 features to 8 features from input data then these 8 optimized features serves as a input of decision tree classifier predict the outcome into two classes. The detailed methodology of the proposed system discussed below.

2.5 SUMMARY

This chapter has reviewed the landscape of existing models used for heart failure prediction, ranging from simpler classifiers like Naïve Bayes to more advanced solutions such as CPT+ and deep learning architectures. While these models have achieved varying levels of accuracy ranging from as low as 60.1% to a maximum of 90.5% they often fall short in other key areas like interpretability, processing speed, and scalability. Through critical analysis, we've identified clear gaps in feature selection, model transparency, and dataset limitations.

Our proposed hybrid model, combining ANN and DT, is designed to address these shortcomings by providing deep learning accuracy with decision tree clarity. By selecting the most relevant features and explaining outcomes in a way that doctors can easily understand, our model aims to go beyond 90% accuracy while remaining efficient and practical. This approach fills a vital gap in current research and moves us one step closer to reliable, real-world heart disease prediction tools.

CHAPTER 3: SYSTEM REQUIREMENT

3.1 SOFTWARE REQUIREMENTS

This section outlines the essential software components needed to develop and deploy the heart failure prediction system. It includes tools for programming, machine learning, data handling, and web interface integration. By using lightweight, widely supported frameworks and libraries, the system remains efficient and accessible. The software stack ensures flexibility for both local and cloud-based execution.

Table 3.1.1 :- List of Software Requirements.

Component	Specification
Operating System	Windows 10 / 11 or Ubuntu 20.04+
Programming Language	Python 3.8 or higher
IDE / Code Editor	VS Code / Jupyter Notebook / PyCharm
ML Libraries	scikit-learn, TensorFlow/Keras, NumPy, pandas
Web Framework	Flask (for backend integration)
Database	Mongodb (for lightweight local data storage)
Visualization Tools	Matplotlib, Seaborn (for graphs, model plots)
Browser Support	Chrome / Firefox / Edge (for frontend UI)
Other Dependencies	pip, virtualenv, SQLite Browser (optional)

3.2 HARDWARE REQUIREMENTS

To support the smooth execution of the system, certain hardware capabilities are required. This section describes the minimum and recommended hardware configurations necessary for training the model, running the web interface, and storing data. It considers both development and real-time prediction scenarios. The aim is to ensure reliable performance without demanding high-end infrastructure.

Table 3.2.1 :- List of Hardware Requirements

Component	**Minimum Requirement**	**Recommended Requirement**
Processor (CPU)	Intel i3 6th Gen or AMD Ryzen 5	Intel i7 10th Gen or AMD Ryzen 7
RAM	8 GB	16 GB or more
Hard Disk Space	250 GB HDD or SSD	500 GB SSD (for faster data handling)
Graphics (GPU)	Integrated Graphics	NVIDIA GTX 1050 Ti or better (for faster ANN training, optional)
Display	1366 x 768 resolution	1920 x 1080 resolution (for UI clarity)
Internet	Required for installing libraries and updates	
Power Backup	UPS or battery backup (recommended)	

CHAPTER 4: METHODOLOGY

4.1 ALGORITHM

To build an accurate heart failure prediction model, we adopted a hybrid approach involving an Artificial Neural Network (ANN) and a Decision Tree (DT). The ANN was used to perform optimized feature selection, extracting essential patterns from the dataset, while the Decision Tree utilized these refined features to perform classification. This section elaborates the step-by-step process followed in the development of the proposed prediction model.

4.1.1 Data Collection

In this study, we used the Cleveland Heart Disease dataset, a standard benchmark for heart-related predictions, obtained from the Kaggle repository. This dataset initially consists of 76 features, but consistent with research practices, only 14 attributes were used, which include 13 independent variables and one dependent (target) variable. These attributes range from demographic details like age and sex to clinical and diagnostic parameters such as chest pain type (cp), resting blood pressure (trestbps), cholesterol (chol), fasting blood sugar (fbs), and maximum heart rate achieved (thalach). The target feature is binary, representing the presence (1) or absence (0) of heart disease. The dataset comprises 1025 patient records with a mix of numerical and categorical data types. Features such as cp, thal, and slope are categorical but encoded numerically, while attributes like age and cholesterol are continuous. This rich mixture allows the model to learn both discrete decisions and continuous patterns. The presence of class imbalance and missing values made this dataset suitable for applying a hybrid technique that can address both data complexity and learning accuracy. It also provides a realistic simulation of clinical data scenarios, making the proposed model highly applicable.

Table 4.1.1 :-

Description of each feature mentioned here and these attributes plays a crucial role for predicting heart failure out of 76 attributes.

Attributes	Description
Age	Patients age in terms of years.
Sex	Patients Gender (male:1,female:0)
Cp	Patient’s chest pain category: 0 for (typical angina), 1 for (atypical angina), 2 for (non- anginal pain), 3 for asymptomatic
Trestbps	The blood pressure level of the patient at rest (in mm/Hg)
Chol	cholesterol content in the blood in mg/dl

Fbs	Blood sugar levels during fasting above 120 mg/dl denoted as 1 for true and 0 for false
restecg	The resting ECG results are classified into three categories : 0 (Normal), 1(ST-T wave abnormality-T wave inversions or ST elevation/depression exceeding 0.05 mV), and 2 (probable/definite left ventricular hypertrophy by Estes criteria).
Thalach	peak heart rate achieved
Exang	Angina induced during exercise 0 indicates NO, 1 indicates Yes
old peak	ST-depression caused by exercise , measured against the resting state
Slope	During peak exercise, the ST segment slope is recorded as : 0- up sloping, 1 – flat, 2- down sloping
Ca	The total number of main vessels (0- 3)
Thal	A blood disorder thalassemia is indicated as : 0- NULL 1- normal blood flow 2- fixed defect (no blood flow in some part of the heart) 3- reversible defect (a blood flow is observed but it is not normal)
Target	target value: 0-no heart failure 1-heart failure.

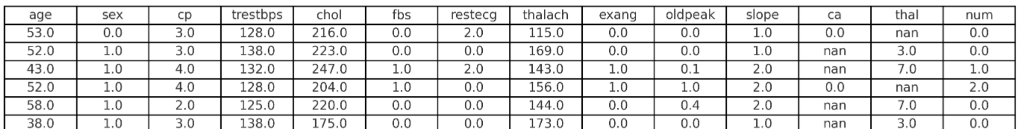
The creators of this dataset listed below.

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D. [8].
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D. [8].
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D. [8].
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D. [8].

4.1.2 Data Preprocessing

Preprocessing was a vital step to ensure clean, consistent, and usable data for model training. The Cleveland dataset contained missing or undefined values particularly in the 'ca' and 'thal' columns. These were addressed through imputation methods like K-nearest neighbors and median replacement based on feature distributions. Categorical attributes such as sex, cp, and thal were encoded using one-hot encoding to preserve their non-ordinal nature. To maintain consistency across features, all numerical attributes were normalized using Min-Max scaling, transforming them into a common range between 0 and 1. This was especially necessary for the ANN model to ensure faster convergence during backpropagation. Outliers and inconsistencies were removed using statistical techniques like z-score analysis. The target variable was also transformed into binary form zero for no disease and one for presence of disease. The preprocessing pipeline was implemented using Python’s pandas and scikit-learn libraries. This cleaned dataset enabled better learning, reduced bias from skewed distributions, and minimized noise. A well-preprocessed dataset improves both model accuracy and generalizability, and it forms the foundation for any reliable machine learning system.

Before training any machine learning model, preprocessing the dataset is a crucial step to ensure data quality and model performance. The Cleveland Heart Disease dataset originally contains 14 attributes, including age, sex, chest pain type, cholesterol levels, and other medical indicators, along with the target label indicating heart disease presence.



age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
53.0	0.0	3.0	128.0	216.0	0.0	2.0	115.0	0.0	0.0	1.0	0.0	nan	0.0
52.0	1.0	3.0	138.0	223.0	0.0	0.0	169.0	0.0	0.0	1.0	nan	3.0	0.0
43.0	1.0	4.0	132.0	247.0	1.0	2.0	143.0	1.0	0.1	2.0	nan	7.0	1.0
52.0	1.0	4.0	128.0	204.0	1.0	0.0	156.0	1.0	1.0	2.0	0.0	nan	2.0
58.0	1.0	2.0	125.0	220.0	0.0	0.0	144.0	0.0	0.4	2.0	nan	7.0	0.0
38.0	1.0	3.0	138.0	175.0	0.0	0.0	173.0	0.0	0.0	1.0	nan	3.0	0.0

Figure 4.1.1:data before preprocessing with noise data

This image illustrates a portion of the dataset **before preprocessing**. As seen, the dataset contains missing values, particularly in the ca and thal columns, where several entries are marked as NaN (Not a Number). These missing values, if left unaddressed, can negatively impact model training by introducing bias or errors.

To handle this, the following preprocessing steps were applied:

1. **Handling Missing Values**: Numerical missing values were either imputed using the column median or removed based on the amount of missing data. For example, missing values in the thal and ca attributes were replaced using the mode (most frequent value) where appropriate.
2. **Data Normalization**: Features such as chol, trestbps, and thalach were normalized using Min-Max scaling to bring all values into a uniform range, improving model convergence.
3. **Label Encoding**: Categorical values such as sex, cp, thal, and slope were encoded into numeric format to ensure compatibility with machine learning models.
4. **Feature Selection**: After preprocessing, the dataset was passed into an Artificial Neural Network (ANN) for dimensionality reduction, where 8 most relevant features were selected for the final model.

After preprocessing, the dataset became cleaner, consistent, and free from null values, forming a solid foundation for reliable and accurate model training. These preprocessing steps significantly improved the overall performance of the hybrid ANN-DT system.

4.1.3 Model Selection

The main objective behind model selection was to combine the strengths of both deep learning and traditional machine learning methods. While traditional classifiers such as Logistic Regression, Naïve Bayes, or K-Nearest Neighbors offer simplicity and interpretability, they often struggle with high-dimensional data and complex non-linear relationships. Conversely, deep learning models such as Artificial Neural Networks (ANNs) can learn intricate feature patterns but lack transparency in decision-making. To strike a balance, we employed a hybrid ANN-DT approach. Here, the ANN serves as a feature extractor, learning compact representations and identifying the most influential attributes. The Decision Tree then uses these extracted patterns to classify heart failure cases. This division of roles ensures both high accuracy and explainability. The ANN compresses raw data into essential patterns, removing redundant or less impactful features. The Decision Tree, being rule-based and transparent, provides easily interpretable decisions. This model combination is effective for medical predictions, where both performance and interpretability are critical. Hence, our model selection rationale is rooted in achieving a synergy between deep learning's learning power and machine learning's clarity.

We have a lot of machine learning algorithms to classify heart disease, but they frequently failed to handle unstructured data. In contrast, supervised learning algorithms can categorize heart disease by using a target variable in the dataset, but this techniques gives best results when dealing with the labeled data. Deep learning algorithms gives a higher accuracy compared to typical learning algorithms and also handle the both structured and unstructured data. There is a wide array of deep learning models and we can select the best one by considering the problem statement and the kind of data we want to handle. we selected ANNs because they can handle structured data effectively, and our dataset consists of numerical and categorical data. To achieve this task, we use hybrid methodologies, ANN is used for selecting optimized features from patient's data then that optimized features are used to categorize heart disease using the decision tree algorithm. Here we mentioned how a single neuron looks like, what are the parts include that.

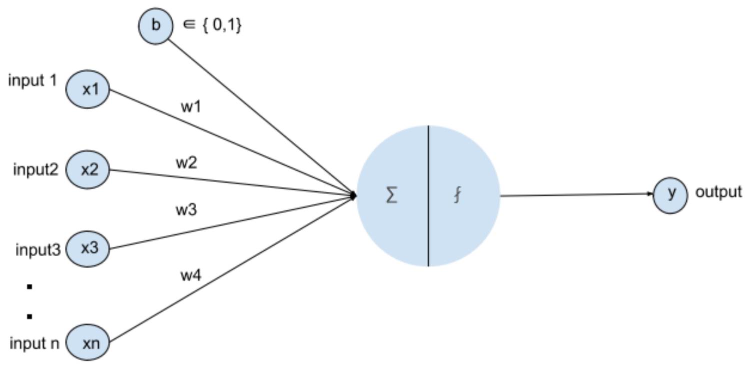


Figure 4.1.2 : Representation of a neuron in a network.

A neuron can process either a single or multiple inputs. In input layer a neuron takes as a single input feature and processed it through transfer function, which calculate the weighted sum of the input along with a bias. The result is then passed into an activation function. The output of each neuron passed to the input of next layers allowing the network to the subsequent layers to process and combine features. Each input is linked to a weight, which determines the its importance and is learned during training, Weights are initialized randomly and adjusted through backpropagation iteratively using the Adam optimizer enables adaptive learning rates for each weight through a stochastic gradient descent approach, combining momentum and RMSProp for faster convergence and stability. In this analysis, the ANN uses Adam optimizer is configured with a learning rate (η) of 0.001 and Binary Crossentropy as the loss function to perform binary classification tasks. The learning rate controls the step size for weight updates, ensuring efficient optimization. Activation functions like ReLU introduce nonlinearity, Allowing the network to learn intricate patterns and propagate information effectively across layers, forming the foundation of neural computations. The mathematical representation of a neuron with weighted sum, bias and activation function as shown below at each layer in network.

$$y = \sum_{i=1}^{n} w_i x_i + b$$

Where,

xi = inputs features passed to the neuron.

wi = Weights are applied to the input.

b= bias term integrated into the weighted sum.

f(.)= activation function.

y=output of the neuron.

As we mentioned earlier to address this problem we were use hybrid approach, By using ANN to extract optimized features, those features would be used for classification of heart disease. In this study we were used decision tree approach for classifying the heart failure because it gives higher accuracies than other traditional techniques. It classify the heart failure based upon the features which can generated by the ANN.

A **Decision Tree** is a supervised machine learning algorithm used for both classification and regression tasks. In the context of heart failure prediction, it plays a key role in classifying whether a patient is at risk or not based on selected features from their medical data. The algorithm mimics human decision-making by creating a **tree-like structure**, where each **internal node represents a condition** (like "Is age > 50?"), each **branch represents an outcome** of the condition, and each **leaf node represents a final decision or class label** (such as “Heart failure: Yes” or “No”).

The decision tree works by splitting the dataset into smaller subsets while building the tree, and this splitting is done using measures like **Gini Index**, **Information Gain**, or **Entropy** to find the best feature to split on. The process continues recursively until the algorithm has broken down the data into the most informative pieces, leading to highly specific classifications.

In your hybrid model, after **ANN selects the top 8 most important features**, these are fed into the Decision Tree. This reduces noise and unnecessary complexity, allowing the DT to perform faster and more accurately. Decision Trees are especially valuable in healthcare because they provide **transparent, interpretable rules**—medical professionals can clearly follow the logic behind a prediction, building trust in the system.

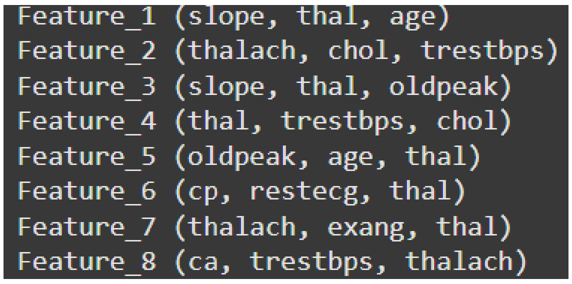
They also require **less data preprocessing** compared to other models and can handle both **numerical and categorical data** with ease. However, they can be prone to **overfitting**, so techniques like **pruning** or combining them in ensemble methods (e.g., Random Forest) are sometimes used. In your case, a standalone Decision Tree works well due to the simplified and refined input provided by ANN, resulting in a powerful, interpretable, and efficient prediction system for heart failure.

4.1.4 Model Training

Training the ANN involved defining a network architecture with one input layer, one hidden layer, and an output layer. The input layer accepts all 14 attributes from the dataset. A dense hidden layer equipped with ReLU activation functions captures non-linear relationships among features. The output layer, responsible for feature selection, compresses the input into 8 optimized features. This layer acts as the penultimate feature extraction stage before classification. The ANN was trained using the Adam optimizer with a learning rate of 0.001. Binary Crossentropy was used as the loss function due to the binary nature of the prediction task. The network was trained for 100 epochs with a batch size of 32. Early stopping was used to prevent overfitting by halting training once validation loss stopped improving. During training, weights and biases were updated iteratively through backpropagation. Once trained, the 8 features from the penultimate layer were extracted and forwarded to the Decision Tree for classification. This training pipeline ensures that the final patterns used by the DT are not arbitrary but derived from a robust deep learning model trained on real patterns in the data.

4.1.5 Feature Extraction

we used an ANN designed to process and optimize the features from the dataset which has 14 attributes. The Ann has multi layers to recognize the most relevant patterns. the weights are adjusted during the training of the Ann which helps to reduce the error in predicting heart failure. higher weights are assigned to features that are mostly contributed to the model performance. The penultimate layer of the ann is used as a feature extractor. This layer simplifies the high dimensional input data into optimized set of features those features help to predict the outcome of heart failure. Based upon certain parameter such as learning rate, step size and optimizer the features patterns may change which can negatively impact on the model efficiency. the dimensionality reduction has done by the penultimate layer not only improves the computational efficiency but minimize the risk of overfitting. This layer compresses the 14 input features into a smaller, optimized set of 8 Patterns. These 8 patterns are a combination of several input features. each pattern or feature represents the higher level learned relationships and interactions that are more meaningful for predicting the target variable. This dimensionality improves the computational efficiency by reducing the feature space from 14 to 8 to make sure the model uses only the most useful features while eliminating the redundancy and noise, once the ANN has generated 8 optimized features(patterns) then they are passed to the decision tree classifier for prediction.



```
Feature_1 (slope, thal, age)
Feature_2 (thalach, chol, trestbps)
Feature_3 (slope, thal, oldpeak)
Feature_4 (thal, trestbps, chol)
Feature_5 (oldpeak, age, thal)
Feature_6 (cp, restecg, thal)
Feature_7 (thalach, exang, thal)
Feature_8 (ca, trestbps, thalach)
```

Figure 4.1.3 : Here we mention the 8 patterns or features, which are combinations of the 14 input features from the dataset, based on the weight updates. The combination of several features to formed a single pattern, for instance Feature_1 pattern is a combination of slope, thal and age features.

4.1.6 Model Evaluation

Evaluate the trained model using the test set to assess its generalization performance.

Use metrics such as accuracy, precision, recall, and F1-score to quantify the model's effectiveness in Heart Failure Prediction System.

4.2 ALGORITHM DESCRIPTION

The proposed hybrid algorithm combines the learning capacity of an Artificial Neural Network with the simplicity and interpretability of a Decision Tree classifier. The process begins with the ANN accepting 14 input features from the preprocessed dataset. The neural network learns weights and biases through training, producing a penultimate layer that holds 8 optimized features a refined representation of the original input. These features are extracted and represent high-value patterns learned through backpropagation and loss minimization. This feature extraction process acts as a dimensionality reduction step, where the ANN filters out irrelevant or redundant information. Each optimized feature may represent a complex combination of original attributes, capturing abstract relationships that are not immediately visible. Once extracted, these 8 features are passed into a Decision Tree model. The DT constructs classification rules by splitting the input data based on feature values that yield the highest information gain. The result is a binary classification: presence or absence of heart failure. This approach combines deep representation learning with interpretable decision paths. The hybrid algorithm improves classification accuracy while maintaining clinical transparency, making it suitable for real-world medical applications.

4.3 SUMMARY

The end-to-end methodology for building the heart failure prediction system involved several structured phases. First, the dataset was collected from Kaggle and loaded into the working environment. Preprocessing was performed to handle missing values in the 'ca' and 'thal' columns using imputation techniques. Categorical features were one-hot encoded, and all numerical features were normalized. After preprocessing, the full dataset consisting of 14 features was fed into an Artificial Neural Network for feature extraction. The ANN was trained with the Adam optimizer and ReLU activation to learn 8 optimized patterns. These patterns represented compressed and meaningful representations of the input features. Next, these 8 features were passed to a Decision Tree classifier. The DT was trained to learn classification rules and generate heart failure predictions. Model performance was evaluated using metrics like accuracy, precision, recall, and F1-score. The ANN achieved 99.71% accuracy, while the hybrid ANN-DT model achieved 98.54% on test data. Finally, the results were visualized, showing classification outputs, confusion matrices, and performance graphs. This complete methodology ensures robustness, accuracy, and clinical usability of the proposed system.

CHAPTER 5: SYSTEM DESIGN

5.1 INTRODUCTION

System design is a crucial phase in the development lifecycle of any project, as it translates the gathered requirements and planned methodology into a structured blueprint. It provides a visual and technical representation of how the system will function, including the flow of data, the interaction between various components, and the responsibilities of different modules. In the context of the **Heart Failure Prediction System**, this chapter focuses on outlining the design architecture using industry-standard tools such as UML (Unified Modeling Language) diagrams and DFDs (Data Flow Diagrams).

The design aims to ensure that the proposed hybrid model using an Artificial Neural Network (ANN) for feature selection and a Decision Tree (DT) for classification is well-integrated into a seamless workflow that allows data collection, processing, prediction, storage, and user interaction via a web interface. These design diagrams serve to illustrate how user inputs are managed, how the model operates in the background, and how results are delivered and stored, ensuring clarity for developers, stakeholders, and evaluators alike.

5.1.1 Objective for Input Design

Input design is focused on capturing accurate, complete, and relevant data from the user in a structured format. The primary objective of input design in this system is to ensure that the doctor can easily provide the necessary patient health parameters such as age, blood pressure, cholesterol level, heart rate, and other relevant factors. The input interface must minimize errors by including validation checks and clear formatting instructions. Well-structured input design not only enhances data integrity but also improves the performance of preprocessing and prediction algorithms. It ensures that all required information is gathered correctly for accurate diagnosis and prediction.

Key objectives include:

- Ensuring completeness and clarity of patient data entry.
- Minimizing user input errors through validations.
- Supporting standardized formats for easy preprocessing.
- Providing a user-friendly interface for smooth data collection.

5.1.2 Output Design

Output design is concerned with how the system communicates results back to the user in this case, the prediction of heart failure risk. The objective of output design is to present the results in a clear, concise, and interpretable manner to assist doctors in making informed decisions. The system's output must be easily understandable, visually structured, and should highlight the risk level along with relevant indicators or confidence scores. Good output design ensures that critical results are not overlooked and that users are guided appropriately based on the outcome.

Key objectives include:

- Displaying heart failure predictions in a meaningful and readable format.
- Providing result summaries with optional insights or recommendation cues.
- Ensuring quick and responsive updates upon processing.
- Enhancing decision-making for doctors through accurate visual output.

5.2 UML DIAGRAMS

5.2.1 Use Case Diagram

The use case diagram for the Heart Failure Prediction System illustrates the primary interactions between the user (doctor) and the system. The main actor in this diagram is the doctor, who interacts with the system to input patient data, initiate prediction processes, and view the results. The doctor first provides relevant patient information such as age, cholesterol level, blood pressure, and other attributes aligned with the Cleveland heart disease dataset. Once the data is entered, the system proceeds to preprocess it, which involves cleaning the data, handling any missing values, and transforming the inputs into a format suitable for the prediction model.

After preprocessing, the system utilizes a hybrid model where an Artificial Neural Network (ANN) is responsible for selecting the most significant features from the input data. These refined features are then passed to a Decision Tree (DT) classifier, which performs the actual prediction of whether the patient is likely to experience heart failure. Following the prediction, the results are displayed to the doctor in an interpretable format, often accompanied by a probability score or classification result (e.g., "At Risk" or "Not at Risk").

The system boundary encapsulates all internal processes, separating the system logic from the user interface. The <<uses>> relationships between the use cases indicate the flow and dependencies among different operations. For example, the process of inputting patient data uses preprocessing, which in turn uses both the prediction module and the result-viewing component. This structured flow ensures that the system processes are carried out in a logical and systematic manner, enhancing usability and reliability for clinical use.

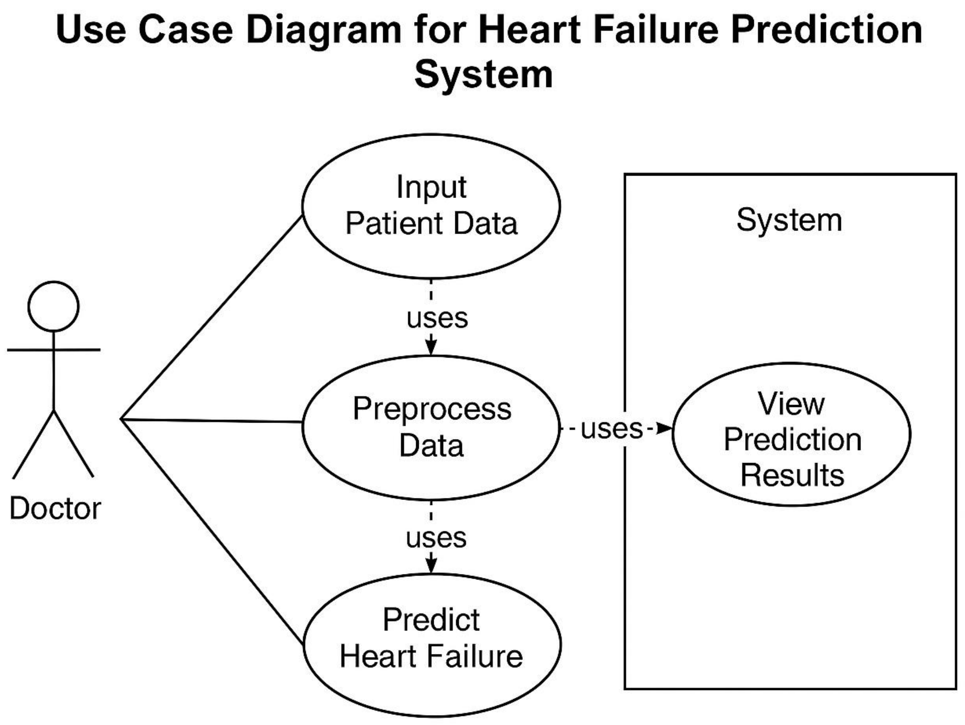


Figure 5.2.1 illustrates the use case diagram for the Heart Failure Prediction System.

- The doctor inputs patient data, which the system preprocesses before running a heart failure prediction.
- The prediction results are then displayed back to the doctor for interpretation.
- The system automates internal processes like preprocessing and classification, while the doctor handles the inputs and views outcomes.

5.2.2 Class Diagram

A **Class Diagram** is one of the key structural diagrams used in UML (Unified Modeling Language) to represent the static structure of a system. It outlines the system's classes, their attributes (data members), methods (functions), and the relationships among various classes. Class diagrams are widely used in object-oriented design to visualize how different objects in a system will interact with each other. This helps developers understand system architecture, maintainability, and scalability before actual implementation. In the context of a heart failure prediction system, the class diagram models how data flows from user input through preprocessing and prediction to final output display.

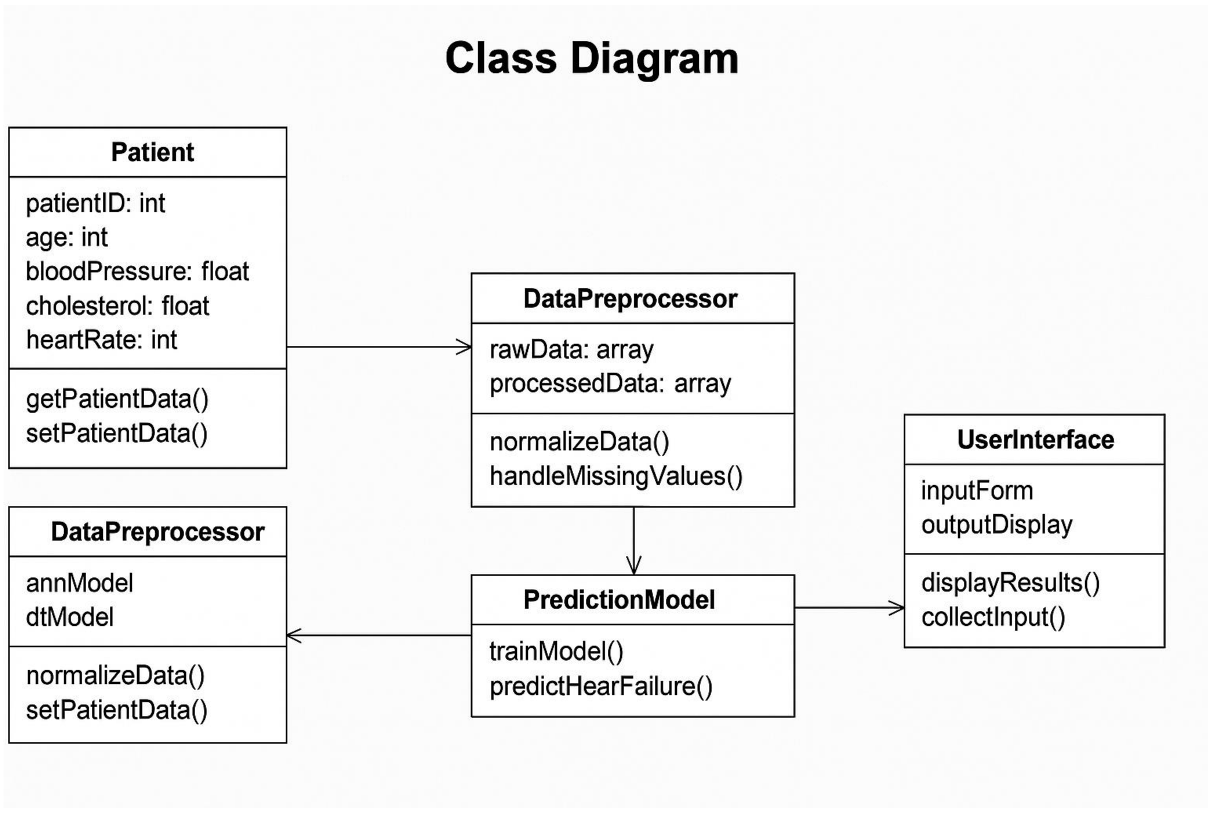


Figure 5.2.2:- Shows the Class Diagram

The class diagram provided above models the key components and interactions within the Heart Failure Prediction System. The central classes involved are Patient, DataPreprocessor, PredictionModel, and UserInterface.

- The Patient class contains attributes such as patientID, age, bloodPressure, cholesterol, and heartRate, which represent basic patient information. It provides two methods: getPatientData() and setPatientData() to retrieve and update patient details.

- The DataPreprocessor class is responsible for cleaning and preparing the input data. It holds rawData and processedData as attributes and includes methods like normalizeData() and handleMissingValues() to standardize the data before it is passed to the prediction model.

- The PredictionModel class handles the machine learning aspect of the system. It includes methods like trainModel() and predictHeartFailure() that are used to build the model and make predictions based on processed input features. This class directly interacts with the DataPreprocessor to get cleaned data for predictions.

- The UserInterface class represents the interaction layer of the system. It manages the inputForm and outputDisplay elements and includes functions like displayResults() and collectInput() to facilitate communication between the user (doctor) and the system.

The relationships shown in the diagram indicate how each component is connected. For example, the Patient class sends data to the DataPreprocessor, which in turn communicates with the PredictionModel. Finally, the PredictionModel sends the outcome to the UserInterface for display. This modular and layered design enhances clarity, maintainability, and logical flow within the system.

5.2.3 Sequence Diagram

A sequence diagram is a type of UML diagram that models the flow of logic within a system in a time-sequenced manner. It shows how different objects interact with each other over time, emphasizing the order in which interactions occur. Sequence diagrams are especially useful for visualizing the dynamic behavior of a system and understanding the communication between various system components in response to specific events.

In the context of the Heart Failure Prediction System, the sequence diagram illustrates the step-by-step interaction between different system entities such as the Doctor, UserInterface, DataProcessor, and PredictionModel during the process of predicting a patient's risk of heart failure.

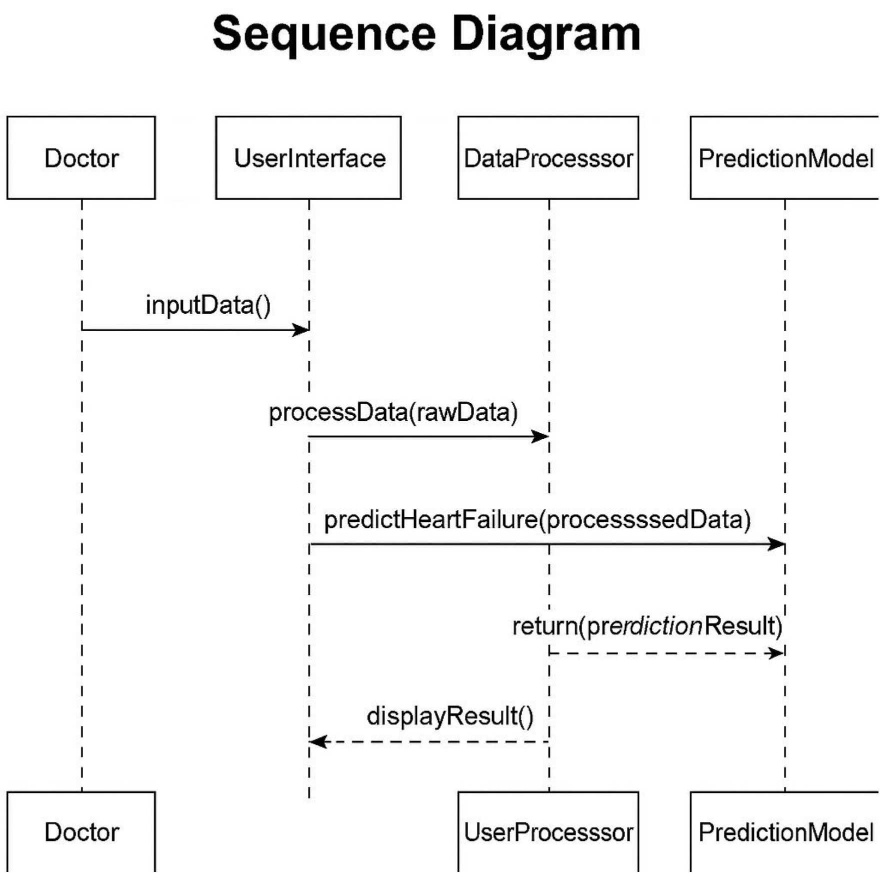


Figure 5.2.3:- Show the Sequence diagram

1. Doctor → UserInterface (inputData()):

   The doctor initiates the interaction by entering patient data through the system’s user interface.

2. UserInterface → DataProcessor (processData(rawData)):

   Once the data is entered, the UserInterface forwards the raw data to the DataProcessor for cleaning, normalization, and handling of missing values.

3. DataProcessor → PredictionModel (predictHeartFailure(processedData)):

   After preprocessing, the cleaned and processed data is passed to the PredictionModel, which contains the logic to analyze the data using the trained ANN-DT model.

4. PredictionModel → DataProcessor (return(predictionResult)):

   The PredictionModel processes the input and returns the prediction result, indicating whether or not the patient is at risk of heart failure.

5. DataProcessor → UserInterface (displayResult()):

   The result is then sent back to the User Interface for display.

6. UserInterface → Doctor:

   Finally, the output (prediction result) is shown to the doctor, enabling them to take appropriate action.

5.2.4 Dfd Diagram

In my system, the Data Flow Diagram (DFD) represents how information flows step by step starting from when a doctor enters patient data, all the way to when the heart failure prediction result is generated and displayed. It helps me visualize the entire process in a clear and organized manner.

It all begins with the Doctor, who is the external user of the system. The doctor inputs raw data such as age, blood pressure, cholesterol, and heart rate. This data first enters the Data Input module (1.0), which is responsible for collecting the values and forwarding them for storage as well as for further processing. The patient data is then stored in the Patient Database, which serves as a reference and also supports the final output generation.

Once the data is entered, it flows into the Data Preprocessing stage (2.0). Here, the system cleans the data, handles missing values, and normalizes it. This step ensures that the information is in the right format for the model to work efficiently. The cleaned and prepared data is then sent to the next phase: Prediction.

The Prediction module (3.0) is the core of my project, where the actual heart failure prediction is carried out. It uses a hybrid ANN-DT model, and it relies on Model Parameters like weights, thresholds, and decision logic, which are fetched from a dedicated parameter store. This allows the prediction to be both accurate and data-driven.

Finally, the results of the prediction are passed to the Result Output module (4.0), where they are structured into a user-friendly format. Alongside the result, relevant parameters and references from the patient database are also used to support the explanation of the result. This output is then shown back to the doctor for final review.

Overall, this DFD helps me ensure that my system is well-structured, each component knows its role, and data flows smoothly from input to output without confusion.

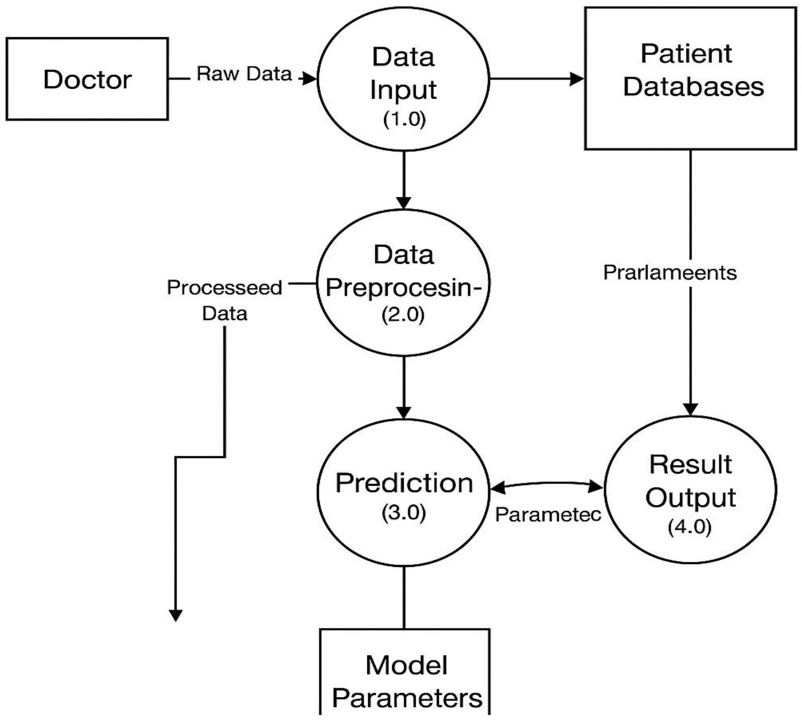


Figure 5.2.4 :- Shows the Data flow diagram with level 1.

5.2.5 Activity Diagram

The Activity Diagram for the Heart Failure Prediction System provides a flowchart-style representation of the system's operations, focusing on the dynamic aspects and the sequence of activities triggered during a prediction session. It highlights control flow between actions and decisions in a step-by-step fashion.

Description:

- The process begins with the doctor initiating the system.
- Patient data is entered via a form.
- The system validates and cleans the data.
- A decision node checks if any values are missing. If yes, it performs data imputation or prompts for re-entry.
- After cleaning, feature selection using the ANN model is conducted.
- The selected features are then passed to the Decision Tree classifier for prediction.
- The classification output is generated.
- The results are formatted and displayed to the doctor.
- The session ends after displaying the result.

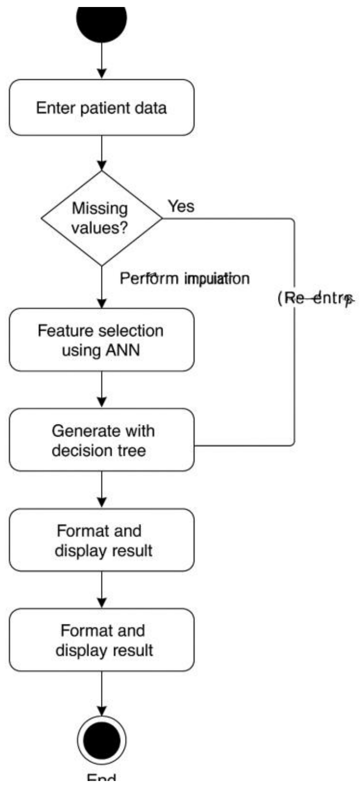

Enter patient data
Missing values?
Yes
Perform impuiation
(Re-entry
Feature selection using ANN
Generate with decision tree
Format and display result
Format and display result
End

Figure 5.2.5 :- Activity Diagram of heart failure prediction system.

It visually represents the step-by-step flow of operations in your system starting from the doctor entering patient data, going through validation, handling missing values, feature selection using ANN, prediction using a Decision Tree, and finally displaying the result.

5.2.6 Component Diagram

The Component Diagram illustrates the high-level structure of the system in terms of its software components and their dependencies. It shows how each module interacts with others during execution.

Key Components:

- **User Interface Component**: Handles patient data input and displays output.
- **Controller Component**: Coordinates workflow, triggering preprocessing and prediction.
- **Preprocessing Module**: Cleans, transforms, and normalizes the data.
- **Feature Selection Module (ANN)**: Selects the most relevant features for classification.

- **Prediction Module (Decision Tree)**: Classifies the patient data into risk categories.
- **Database Component**: Stores patient data and model parameters.
- **Result Formatter**: Converts raw results into an interpretable format for the doctor.

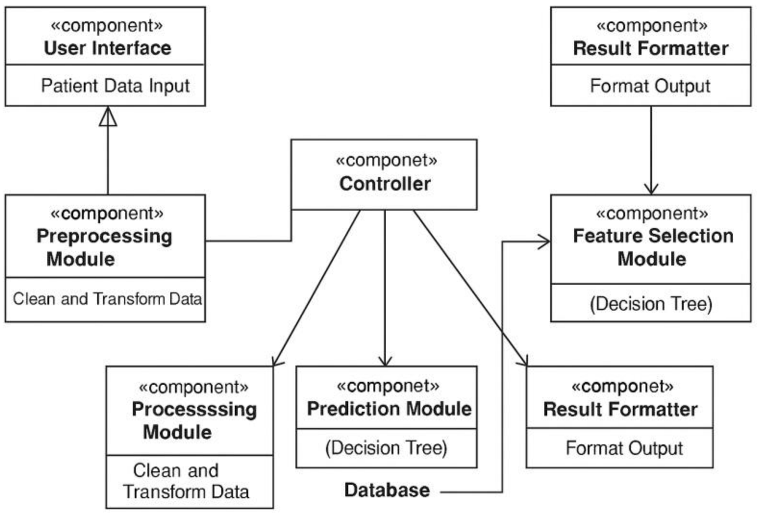


Figure 5.2.6 :- Component diagram of heart failure prediction system

Breaks the system into modular components like UI, Controller, Model, and Database.Shows dependencies among components. Useful for understanding system modularity and organization.

5.2.7 Deployment Diagram

The Deployment Diagram represents the physical architecture of the system, including where software components are deployed across hardware nodes.

Structure:

- Client Device (Doctor’s System): Runs the frontend (User Interface). May be a browser or desktop app.
- Web Server: Hosts the backend logic — controller, preprocessing, prediction engine.
- Model Server: Stores the trained ANN and Decision Tree models.

- Database Server: SQLite or other lightweight database storing patient records and system logs.

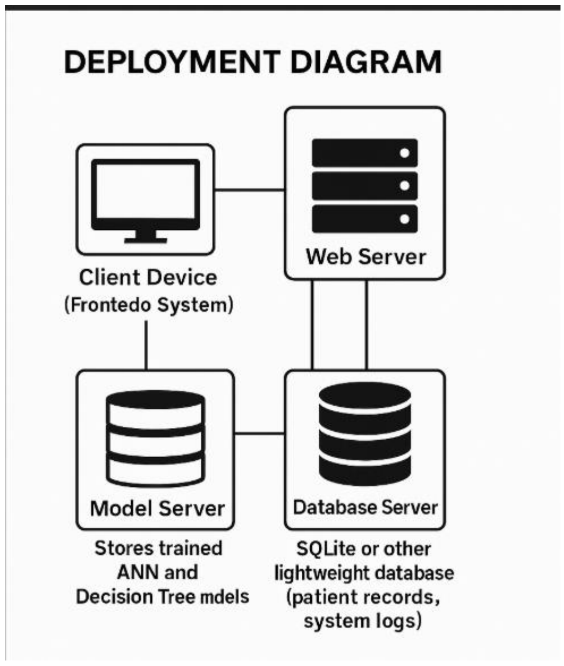

DEPLOYMENT DIAGRAM
Client Device
(Frontedo System)
Web Server
Model Server
Stores trained ANN and Decision Tree mdels
Database Server
SQLite or other lightweight database (patient records, system logs)

Figure 5.2.7 : Deployment Diagram of the Heart Failure Prediction System

5.2.8 State Diagram

The State Diagram tracks the lifecycle of a prediction request within the system, outlining the various states the system enters as it processes patient data.

States and Transitions:

- **Idle**: Waiting for doctor input.
- **Data Entry**: Receiving patient data.
- **Preprocessing**: Cleaning and transforming the data.
- **Feature Selection**: Using ANN to reduce dimensionality.
- **Classification**: Using DT for prediction.
- **Result Display**: Showing output to the doctor.
- **Session Complete**: Process ends or restarts for a new patient.

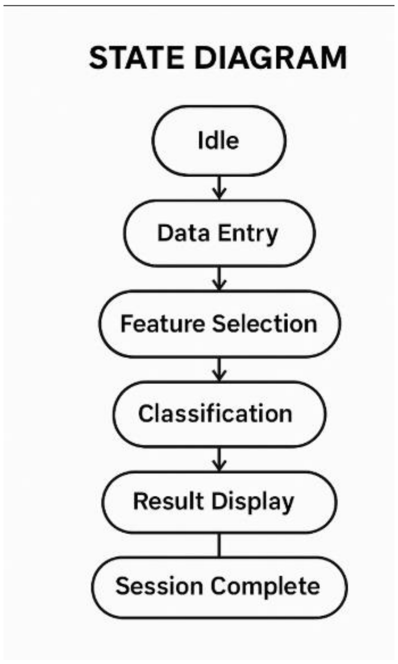


Figure 5.2.8 : State Diagram of the Heart Failure Prediction System

Represents the lifecycle of a prediction request.Includes states like Idle, Data Entry, Preprocessing, Classification, and Result Display.Tracks how the system transitions from one state to another.

CHAPTER 6: SYSTEM STUDY AND TESTING

In this chapter, I've focused on the complete analysis and evaluation of how my heart failure prediction system functions in real-time. The aim of this chapter is to study the different modules involved in the system, understand how they interact, and assess the accuracy, reliability, and performance of the system through proper testing.

The System Study portion explains each core module from the user's point of view starting from login, data input, model interaction, to final result visualization. I've broken down each page or screen the user interacts with and described its role in the overall flow.

The Testing section highlights the methods I used to validate the system's performance. This includes functionality testing, user input handling, output correctness, and prediction accuracy. I also made sure to verify that all components of the system from frontend to backend, including the SQLite database and the hybrid ANN-DT model work seamlessly together.

Ultimately, this chapter reflects the transition from development to real-world usability, ensuring that the heart failure prediction system is both efficient and trustworthy for medical use

6.1 USER MODULES

The User Module is the front-facing part of my heart failure prediction system, designed to provide a smooth and interactive experience for both doctors and users. It includes essential features like login, input data submission, viewing predictions, and tracking model scores. Each sub-module ensures user-friendly navigation and accurate interaction with the backend prediction engine.

6.1.1 View Home Page

The Home Page of my heart failure prediction system is designed to be simple, user-friendly, and welcoming. It serves as the entry point to the system, offering both login and sign-up options. The layout is divided into two sections:

- Left Panel – Sign In:

  This section is meant for existing users such as doctors or medical professionals who already have an account. They can sign in by entering their email and password, or alternatively, choose from four popular social media login options:

  - Google
  - Facebook
  - GitHub

- LinkedIn

This allows flexibility and convenience for users who prefer using their social accounts for authentication. There's also a "Forgot Your Password?" link in case users need to recover access.

- Right Panel – Sign Up Prompt:

This part is more colorful and visually inviting, with a "Welcome, Friend!" message encouraging new users to register. The SIGN UP button leads to the registration page where users can create an account to access the system features.

6.1.2 Registration – Create Account

In this module, users (typically doctors or medical staff) can register themselves to access the heart failure prediction system. The registration page is cleanly designed with a dual-panel layout. On the right side, the user is prompted to **Create an Account** by entering the following details:

- **Name** – Full name of the user.
- **Mobile Number** – A valid contact number.
- **Email** – A working email address that will serve as the login ID.
- **New Password** – A strong password for account security.
- **Confirm Password** – To verify the entered password.

Once all fields are filled, clicking the **SIGN UP** button submits the data to the backend. The system checks for any existing accounts with the same email, validates password confirmation, and securely stores user credentials. On successful registration, the user is redirected to the login page with a success message.

6.1.3 Input Form

In this section, the user (usually a doctor or healthcare professional) is presented with an interface to input patient details required for heart failure prediction. This form is built to match the attributes used in the Cleveland Heart Disease dataset, ensuring that the input data aligns with the trained model's expectations.

The input fields typically include 13 medical attributes such as:

- Age

- Sex
- Chest Pain Type
- Resting Blood Pressure
- Cholesterol
- Fasting Blood Sugar
- Rest ECG
- Maximum Heart Rate Achieved
- Exercise Induced Angina
- ST Depression
- Slope of ST Segment
- Number of Major Vessels
- Thalassemia

Once the user fills out the patient details and submits the form, the input is preprocessed in the backend and passed to the hybrid prediction model (ANN for feature selection and Decision Tree for classification). The predicted result is then stored in the database and shown to the user on the result page.

6.1.4 View Results

After submitting the patient’s data through the input form, the system processes the input using the hybrid ANN-DT model and generates a prediction. In the **View Results** section, users can see the outcome of the prediction clearly displayed.

This page typically shows:

- The predicted result (e.g., **"Heart Disease Detected"** or **"No Heart Disease"**)
- Basic patient information that was submitted
- A timestamp of when the prediction was made

This feature is designed for clarity and ease of use, helping doctors quickly interpret whether the patient is at risk of heart disease based on the model’s decision. The results are also stored in the mongodb database for record-keeping and future reference.

6.2 SYSTEM MODULES

6.2.1 Working on Dataset

This stage involves importing and understanding the Cleveland Heart Disease dataset. The data is examined for completeness, correctness, and suitability. Irrelevant or redundant data is identified and marked for removal or correction.

6.2.2 Pre-Processing

Here, raw data is cleaned and transformed. Null values are handled, categorical data is encoded, and features are scaled. This step ensures the dataset is in a proper format for machine learning models.

6.2.3 Training the Data

The dataset is divided into training and testing sets. The training set is fed into the Artificial Neural Network (ANN) model to learn patterns, using ReLU activation and Adam optimizer.

6.2.4 Model Building

This involves constructing the hybrid model, where ANN selects the best features, and a Decision Tree classifier uses those features for classification. The architecture is tuned to enhance prediction accuracy.

6.2.5 Generated Score

Once trained, the model generates a performance score (accuracy, precision, recall, etc.). These scores help in evaluating the effectiveness of the model.

6.2.6 Generate Results

Using the finalized model, predictions are made on test data or new inputs. The output is displayed and stored in the database, giving meaningful insights about the presence or absence of heart failure.

6.3 SYSTEM STUDY ANALYSIS

6.3.1 Feasibility Study

Evaluates whether the system is practical and achievable. It checks all technical, economic, and social aspects to confirm that the project can be successfully completed.

6.3.2 Economical Feasibility

Analyzes cost-related aspects. Since open-source tools like Python and mongodb are used, the overall cost is minimal, making the project economically viable.

6.3.3 Technical Feasibility

Examines the tools and technologies used. The system uses Python (Flask for web), mongodb (for database), and machine learning libraries like TensorFlow or Scikit-learn, all of which are technically feasible.

6.3.4 Social Feasibility

Assesses the impact of the system on users and society. Since it assists in early heart failure detection, it is beneficial and socially acceptable.

6.4 SYSTEM TESTING

6.4.1 Unit Testing

Tests individual components like the input form, prediction logic, and database entries separately to ensure they work as expected.

6.4.2 Integration Testing

Verifies the interaction between modules (e.g., how user input flows into the model and returns the output), ensuring seamless integration.

6.4.3 Acceptance Testing

Validates whether the system meets user requirements. Doctors or testers review the output to confirm the system behaves correctly in real-world conditions.

6.4.4 Functional Testing

Ensures that all features (registration, login, input, output, score view, etc.) work as specified in the functional requirements.

6.4.5 White Box Testing

Involves testing the internal structure of the application. This includes verifying functions, logic flow, and performance of individual modules.

6.4.6 Black Box Testing

Focuses on system outputs based on various inputs without considering the internal code. This helps simulate real-world user interaction.

6.5 TEST OBJECTIVES

This section defines the purpose of testing: to identify bugs, validate performance, ensure data security, verify usability, and confirm that the application meets all user needs and project specifications.

6.6 FEATURES TO BE TESTED

Lists all the core features that need testing, such as:

- User registration/login
- Input model form
- Result generation
- Score visualization
- Data saving to Mongodb
- Frontend-backend integration

CHAPTER 7: TOOLS AND LIBRARIES

7.1 INTRODUCTION

This chapter introduces the core tools and libraries that form the technological foundation of your heart disease prediction project. In developing a machine learning application integrated with a Flask web interface, a number of software components play significant roles from model training to data handling and real-time prediction delivery. Python serves as the central language, enabling seamless communication between machine learning models, web infrastructure, and data storage. Libraries such as TensorFlow and Keras facilitate deep learning model development, while Scikit-learn aids in preprocessing and traditional machine learning tasks. MongoDB functions as a reliable storage solution for structured data, capturing both input and prediction results. Additionally, tools for data visualization like Matplotlib and Seaborn are instrumental in exploring data and evaluating model performance. The use of Jupyter Notebook supports an iterative development environment for model prototyping. Altogether, these tools provide a robust, end-to-end framework that supports both the development and deployment phases of your intelligent heart disease prediction system.

7.2 PYTHON PROGRAMMING LANGUAGE

Python is the central programming language that connects every component of your project. Its clarity, simplicity, and versatility make it particularly suitable for both machine learning development and web application deployment. Your Flask-based web application uses Python to manage user interactions, handle HTTP requests, and interface with the backend models and database. Additionally, the Python environment is used extensively in developing the machine learning pipeline, from data loading and preprocessing to model training and evaluation. The language's ability to integrate various third-party libraries allows for a smooth and modular implementation. Libraries are seamlessly imported and utilized for specific tasks, creating a structured and maintainable codebase. Python's wide adoption in the data science community ensures ongoing support, frequent updates, and a vast pool of reusable resources, making it the ideal choice for building your heart disease prediction system.

7.3 TENSORFLOW AND KERAS

TensorFlow and Keras are at the core of the deep learning component in your system, specifically in training the Artificial Neural Network used for feature extraction. TensorFlow provides a robust backend framework for defining and executing computational graphs, while Keras, as a high-level API, simplifies the process of building and training models. You employ a sequential model that consists of multiple dense layers designed to learn complex patterns within the data. This model is optimized using advanced techniques such as adaptive gradient descent and is trained on normalized data to improve generalization. After training, the model is saved and later integrated into your Flask web application. There, it is loaded and used to process new user inputs, extracting meaningful features that will feed into the decision-making system. This demonstrates TensorFlow's scalability and performance capabilities and Keras' ease of use, both of which are essential for real-world machine learning applications like yours.

7.4 SCIKIT-LEARN

Scikit-learn plays a crucial role in the machine learning pipeline, particularly in preprocessing and classification using traditional algorithms. In your project, Scikit-learn is used to standardize features using tools like the StandardScaler, ensuring that the data fed into the models is normalized and suitable for training. More importantly, it is utilized to train the Decision Tree classifier, which operates on the features extracted by the ANN. This classifier interprets the features and makes the final prediction regarding heart disease risk. Scikit-learn also provides a suite of evaluation metrics such as accuracy and confusion matrix, allowing you to assess model performance quantitatively. Its utility extends into the deployment phase, where the trained scaler and classifier are loaded into the Flask application for real-time predictions. Scikit-learn thus serves as an indispensable tool for integrating classical machine learning techniques into your modern, hybrid prediction system.

7.5 PANDAS

Pandas serves as the primary tool for data handling and manipulation during the early stages of model development. It is used extensively to load the dataset from CSV files, extract relevant features and labels, and structure the data in a format conducive to analysis and modeling. Its DataFrame structure allows for intuitive data slicing, filtering, and transformation, which significantly streamlines the preprocessing phase. Pandas enables efficient operations on tabular data, such as identifying missing values, encoding categorical variables, or summarizing key statistics. Although it is not directly used in the deployed Flask application, its contribution is foundational in shaping the dataset into a form that can be effectively used by the ANN and decision tree models. Without this step, the subsequent stages of scaling, modeling, and evaluation would not function as effectively, underscoring Pandas' importance in the development pipeline.

7.6 NUMPY

NumPy underpins the numerical computations required for both model training and real-time inference. It enables efficient array-based operations, which are essential when dealing with high-dimensional data and mathematical computations in machine learning. During the preprocessing phase, data extracted using Pandas is converted into NumPy arrays to facilitate faster and more efficient operations. These arrays are then fed into the models for training and prediction. In the Flask web application, NumPy is instrumental in converting user inputs into the correct format expected by the scaler and model. It handles array transformations and numerical computations that are required before passing the data through the trained ANN and Decision Tree models. Its role may seem subtle compared to other libraries, but NumPy provides the performance backbone that ensures computations are fast, stable, and consistent across the system.

7.7 MATPLOTLIB AND SEABORN

Visualization is key to understanding both the data and the model's performance, and this is where Matplotlib and Seaborn come into play. During the model development phase, these libraries are used to create visual representations of training history, such as graphs showing accuracy and loss over epochs. They also help in visualizing the importance of different features, which is crucial in validating the relevance of attributes selected by the ANN. Correlation heatmaps generated using Seaborn allow for an intuitive understanding of how different features relate to each other, helping to guide feature selection and engineering decisions. While these visualizations are not part of the Flask deployment, they provide valuable insights during the analysis and research stages. They help ensure that your models are not just accurate but also interpretable and trustworthy, especially when applied to a critical domain like heart disease prediction.

7.8 JUPYTER NOTEBOOK

Jupyter Notebook offers a flexible, interactive environment for developing and testing your machine learning models. Although your final implementation is structured as Python scripts integrated into a Flask application, much of the early experimentation, debugging, and visualization likely occurred within a Jupyter Notebook. The notebook's cell-based execution allows you to test individual components such as data loading, model architecture, or training routines without executing the entire script. This feature is invaluable during model tuning, as it enables quick iteration and immediate feedback. Moreover, the ability to integrate rich text, plots, and code in a single interface makes it easier to document the development process and share findings. Jupyter serves as the ideal platform for prototyping before transitioning to a production-level implementation in your heart disease prediction system.

7.9 SUMMARY

To summarize, your heart disease prediction system is built on a carefully chosen stack of tools and libraries that work in harmony to deliver a robust, scalable, and interpretable application. Python serves as the core language, facilitating smooth integration between machine learning models, web infrastructure, and data storage. TensorFlow and Keras drive the deep learning model, while Scikit-learn contributes to preprocessing and traditional classification. Pandas and NumPy handle data manipulation and numerical operations, ensuring the data pipeline is efficient and consistent. Visualization tools like Matplotlib and Seaborn enhance understanding and transparency during the model development process. MongoDB enables structured data storage, and Flask serves as the bridge between users and the model through an interactive web interface. Finally, Jupyter Notebook provides a development environment ideal for prototyping and experimentation. Together, these tools empower your system to function as a practical, real-world application in the domain of heart disease prediction.

CHAPTER 8 :- RESULTS & DISCUSSIONS

8.1 RESULTS DEMONSTRATION

8.1.1 Registration page

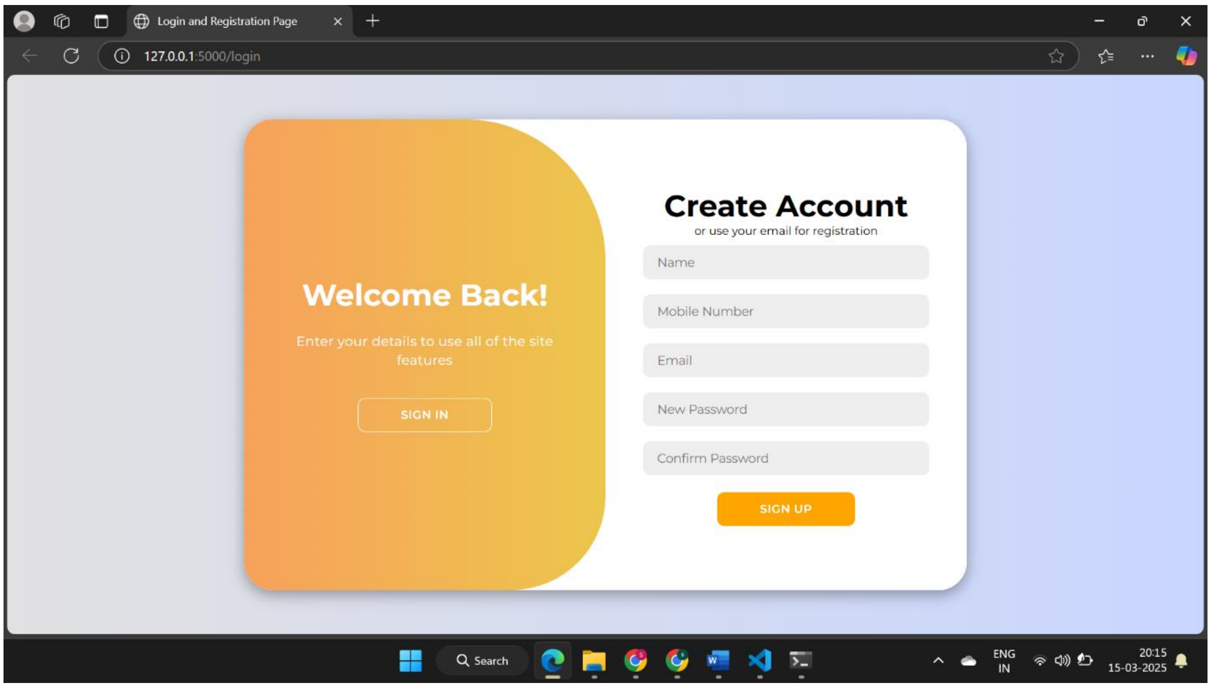

Login and Registration Page
127.0.0.1:5000/login
Welcome Back!
Enter your details to use all of the site features
SIGN IN
Create Account
or use your email for registration
Name
Mobile Number
Email
New Password
Confirm Password
SIGN UP

Figure 8.1.1 : The registration page allows users to sign up for an account, enabling access to the heart disease prediction interface

The registration page is the entry point for new users to create an account. It typically includes fields like name, email, password, and confirmation details to ensure secure sign-up. Once registered, users gain access to the heart disease prediction interface. This interface uses user-provided health data to assess potential heart disease risk. The registration process ensures data privacy and personalized access. It also helps maintain secure and trackable usage of the prediction tool.

8.1.2 Home Page

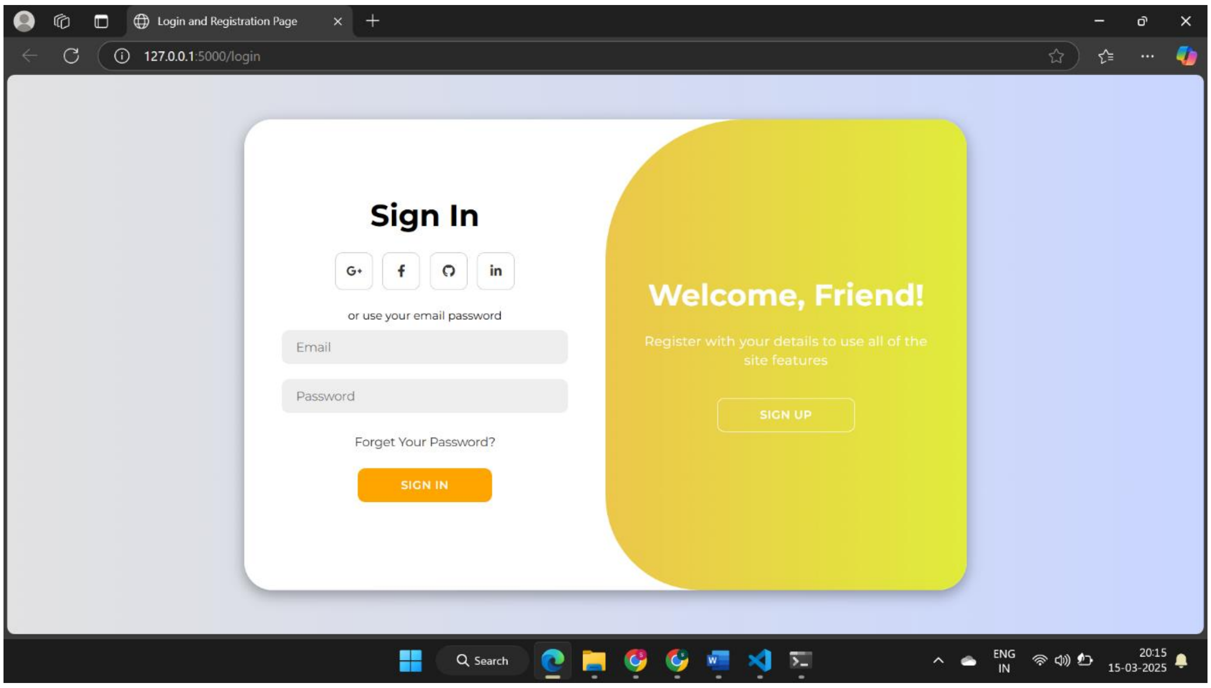
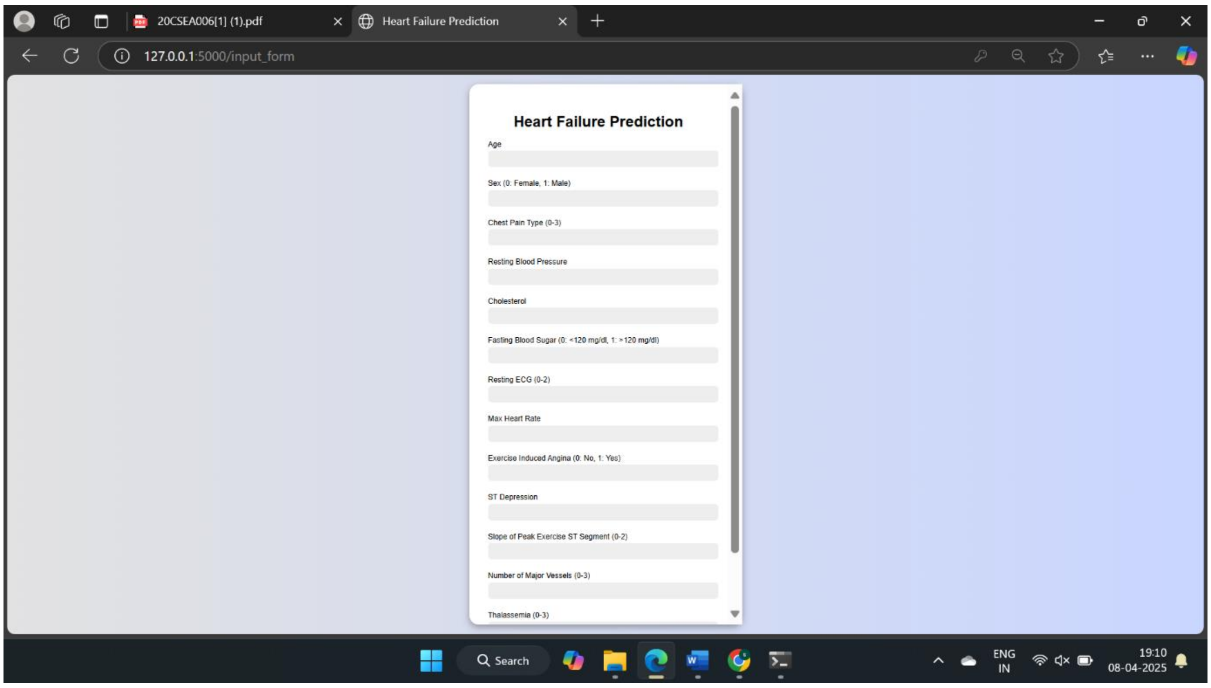

Login and Registration Page
127.0.0.1:5000/login
Sign In
or use your email password
Email
Password
Forget Your Password?
SIGN IN
Welcome, Friend!
Register with your details to use all of the site features
SIGN UP

Figure 8.1.2 : The home page of the heart disease prediction project, displaying the initial login interface.

The registration page of the heart disease prediction project provides a secure platform for new users to create an account. It includes input fields for essential user details such as username, email, and password. This step ensures that only registered users can access the system’s features. Upon successful registration, users are redirected to the login interface. The process helps maintain user-specific records and supports personalized interaction with the prediction system. It also enhances the overall security and management of user data.

8.1.3 Input Form



Heart Failure Prediction

Age

Sex (0: Female, 1: Male)

Chest Pain Type (0-3)

Resting Blood Pressure

Cholesterol

Fasting Blood Sugar (0: <120 mg/dl, 1: >120 mg/dl)

Resting ECG (0-2)

Max Heart Rate

Exercise Induced Angina (0: No, 1: Yes)

ST Depression

Slope of Peak Exercise ST Segment (0-2)

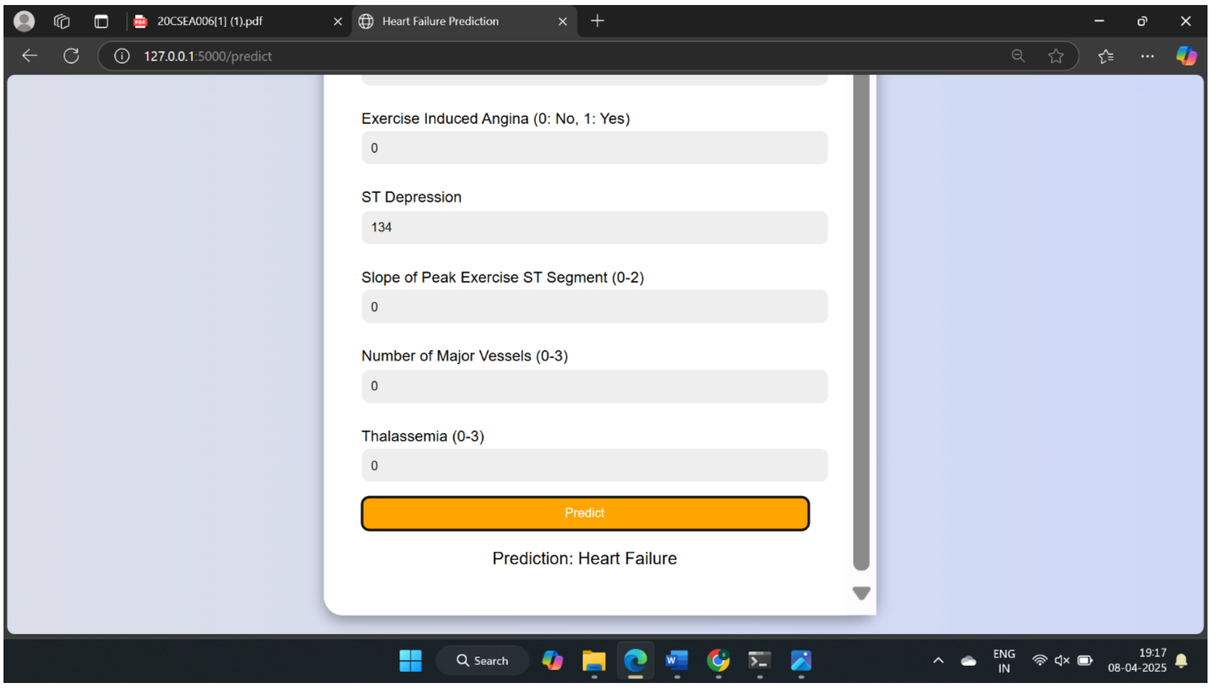
Number of Major Vessels (0-3)

Thalassemia (0-3)

Figure 8.1.3 : The input form page where users enter health data for heart disease prediction.

The registration page is the first step for users to access the heart disease prediction system. It allows new users to create an account by providing basic details like username, email, and password. Once registered, users can log in and access the input form page. This form collects 13 crucial health attributes: age, sex, chest pain type (cp), resting blood pressure (trestbps), serum cholesterol (chol), fasting blood sugar (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression (oldpeak), slope of the ST segment (slope), number of major vessels colored by fluoroscopy (ca), and thalassemia (thal). These inputs are processed by the model to predict the likelihood of heart disease. The registration ensures secure access and personalized usage of the prediction system.

8.1.4 Results



Exercise Induced Angina (0: No, 1: Yes)
0

ST Depression
134

Slope of Peak Exercise ST Segment (0-2)
0

Number of Major Vessels (0-3)
0

Thalassemia (0-3)
0

Prediction: Heart Failure

Figure 8.1.4 : This figure displays the prediction output in text format, indicating the heart disease diagnosis.

Users are required to enter details such as username, email, and password to create an account. After registration, they are directed to the input form page, where they can provide 13 health-related attributes including age, sex, chest pain type, blood pressure, cholesterol, and more. These inputs are essential for the system to predict the presence or absence of heart disease.

8.1.5 Visualization

The feature importance graph illustrates the contribution of each extracted feature in the Decision Tree model for heart failure prediction.The graph effectively highlights the hierarchical importance of features, showcasing how specific combinations of clinical parameters enhance the model's decision making capability during the feature extraction process.

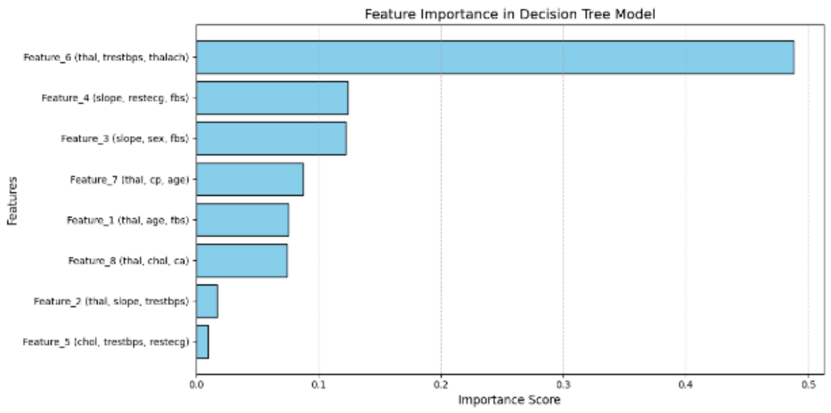
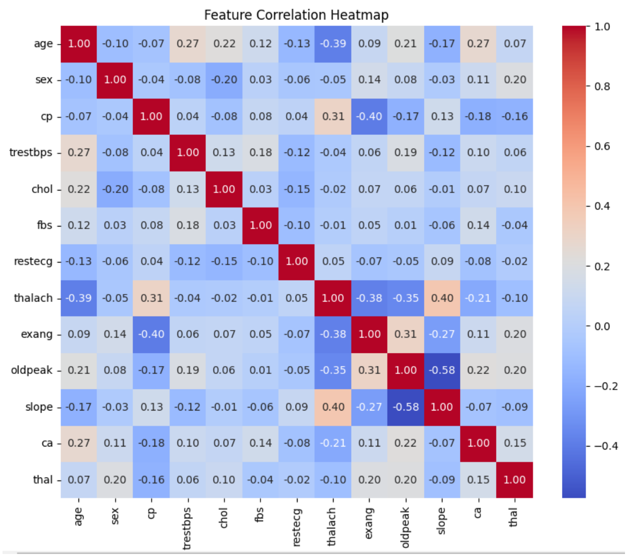


Figure 8.1.5 : Feature Importance in decision tree

In the above figure shows the Feature_6(thal, trestbps,thalach) plays a major role in decision tree to classify the heart failure,Feature_4(slope,restecg,fbs) and Feature_3(slope,s ex,fbs) plays a second-highestimportance followed by Feature_7(thal,cp,age),Feature_1(thal,age,fbs),Feature_6(thal,chol,ca),Feature_2(thal,slope,tr estbps)an d Feature_5(chol,trestbps,restecg) which plays a minor roles for predicting the heart failure in view of decision tree.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
age	1.00	-0.10	-0.07	0.27	0.22	0.12	-0.13	-0.39	0.09	0.21	-0.17	0.27	0.07
sex	-0.10	1.00	-0.04	-0.08	-0.20	0.03	-0.06	-0.05	0.14	0.08	-0.03	0.11	0.20
cp	-0.07	-0.04	1.00	0.04	-0.08	0.08	0.04	0.31	-0.40	-0.17	0.13	-0.18	-0.16
trestbps	0.27	-0.08	0.04	1.00	0.13	0.18	-0.12	-0.04	0.06	0.19	-0.12	0.10	0.06
chol	0.22	-0.20	-0.08	0.13	1.00	0.03	-0.15	-0.02	0.07	0.06	-0.01	0.07	0.10
fbs	0.12	0.03	0.08	0.18	0.03	1.00	-0.10	-0.01	0.05	0.01	-0.06	0.14	-0.04
restecg	-0.13	-0.06	0.04	-0.12	-0.15	-0.10	1.00	0.05	-0.07	-0.05	0.09	-0.08	-0.02
thalach	-0.39	-0.05	0.31	-0.04	-0.02	-0.01	0.05	1.00	-0.38	-0.35	0.40	-0.21	-0.10
exang	0.09	0.14	-0.40	0.06	0.07	0.05	-0.07	-0.38	1.00	0.31	-0.27	0.11	0.20
oldpeak	0.21	0.08	-0.17	0.19	0.06	0.01	-0.05	-0.35	0.31	1.00	-0.58	0.22	0.20
slope	-0.17	-0.03	0.13	-0.12	-0.01	-0.06	0.09	0.40	-0.27	-0.58	1.00	-0.07	-0.09
ca	0.27	0.11	-0.18	0.10	0.07	0.14	-0.08	-0.21	0.11	0.22	-0.07	1.00	0.15
thal	0.07	0.20	-0.16	0.06	0.10	-0.04	-0.02	-0.10	0.20	0.20	-0.09	0.15	1.00

Figure 8.1.6: correlation heatmap shows the relationship between the features

The correlation heatmap helps us to find out how different features are related to each other. By observing it, we can identify features which are very similar and which can negatively impact on our model's performance. By removing unnecessary features, we can keep only the most important ones which provides the most value. This helps us to simplify our data and improve our model's speed and accuracy By removing unnecessary features, we can keep only the most important ones which provides the most value. This helps us to simplify our data and improve our model's speed and accuracy.

8.2 ACCURACY RESULTS

8.2.1 Performance of ANN

The proposed model employs a type of feedforward neural network, specifically referred to as a Multi Layer Perceptron (MLP), for feature extraction. The feedforward architecture ensures that data flow in one direction, from the input layer to output layer through hidden layer.

Network Architecture

✓ Network Type: Multi-Layer Perceptron.

✓ Total Layers: 3.

✓ Input Layer: Receives 14 attributes from dataset as input features.

✓ Hidden Layer: Includes neurons with ReLU activation function to introduce nonlinearity and learn complex patterns.

✓ Output Layer: Extracts 8 optimized features from the penultimate layer.

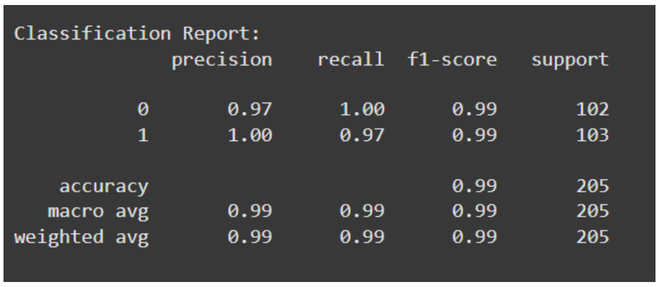
The network architecture uses the ReLU (Rectified Linear Unit) activation function for the hidden layers, as it has been shown to deliver the maximum average accuracy in comparison to other activation functions. This conclusion aligns with the findings from the base paper, where the authors demonstrated that ReLU performs exceptionally well due to its ability to efficiently handle nonlinearity and avoid the vanishing gradient problem. Based on these insights, ReLU was chosen as the activation function across all hidden layers in this study.

8.2.2 Testing and Overall Accuracy of ANN

After training our ANN, we tested it on a separate data set to see how well it was working. And we utilized 80% of the data for train the model, while 20% was set aside for testing. The results indicated that our ANN was very accurate, by getting the testing accuracy of 98.54%. Where this means its effectiveness in learning and identifying important patterns in the data. So overall, our ANN was able to predict heart disease outcomes with an accuracy of 99.71%. This determines that our model is robust, efficient, and effective in making predictions.

8.2.3 Decision Tree Performance

After selecting the optimized features using the ANN, we used the Decision Tree classifier to predict heart disease cases. We tested this model using a separate dataset, and we got the best results. The Decision Tree classifier was more accurate in identifying the people with or without heart disease, with very few mistakes. The results determined high precision and recall, with an overall accuracy of 98.54%. This confirms that the Decision Tree classifier works well with the optimized features selected by the ANN to make the best predictions.



```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       102
           1       1.00      0.97      0.99       103

    accuracy                           0.99       205
   macro avg       0.99      0.99      0.99       205
weighted avg       0.99      0.99      0.99       205
```

Figure 8.2.1 : Here we mentioned the detail classification report of decision tree with their performance.

To assess the performance of my model, I conducted a detailed analysis of its predictions on the test set. The results show a high degree of accuracy, with most predictions being correctly classified. This suggests that the feature extraction and selection process I followed has contributed effectively to the model performance. The detailed breakdown of accuracy, along with other relevant metrics, provides a clear view of how well the model has generalized to the test data. Further details can be seen in the image below.

8.2.4 Detailed Accuracy Breakdown

Total Test Samples: 205

Correct Predictions: 202

Incorrect Predictions: 3

Decision tree Accuracy : 98.54%

The Detailed Accuracy Breakdown signifies that the decision tree predicts the correct outcome ,when tested on 20% of dataset, After completing the testing on 205 samples , the decision tree model correctly predicts 202 rows and falsely predicts 3 sample out of 205 sample. It means that if the entire dataset tested which consists of 1025 rows the decision tree classifier correctly predicts 1010 samples out of 1025 samples.

8.2.5 Key Findings

The hybrid ANN-DT model achieved remarkable results : The ANN effectively extracted 8 optimized features or patterns from the original dataset, achieving a testing accuracy of 98.54% and an overall accuracy of 99.71%. The Decision Tree classifier successfully utilized these features, achieving a testing accuracy of 98.54 % and exhibiting high precision and recall.

8.3 Discussion of Results with Existing Systems

To better understand the strengths and limitations of existing heart failure prediction approaches, we have compiled a set of comparative tables. These tables summarize and contrast various aspects of previous studies, including the models employed, the features considered, the datasets used, and overall performance metrics. This comparative analysis highlights key patterns, identifies existing gaps, and justifies the need for a hybrid ANN-DT framework. By positioning our proposed approach alongside these existing systems, we demonstrate its potential advantages in terms of accuracy, interpretability, and robustness.

8.3.1 Model Comparison

Table 8.3.1 :- Model Comparison with Existing Model and their remarks.

Study	Model Used	Accuracy (%)	Remarks
Mahgoub [1]	Deep Learning (CPT+)	90.5	High accuracy, but slow on large datasets
Priyanga [5]	Naïve Bayes (NBwa)	72.91	Lightweight but lacks complex understanding
Agrawal [6]	Random Forest + EDA	79.1	Balanced model, limited optimization
Marbaniang [9]	Multiple ML algorithms	85.49	ANN-based approach, decent results
Saravanan [7]	K-Means + SVM	86.3	Good hybrid, lacks deep learning strengths
Proposed Model	**ANN + Decision Tree**	**98.54**	**High accuracy, interpretable, fast**

This table provides a detailed comparison of different models used in recent studies for heart disease prediction, focusing on their accuracy and key observations. Mahgoub [1] employed a deep learning model called CPT+, which delivered a high accuracy of 90.5%. However, this method is computationally intensive and becomes slower when handling large datasets, which may affect its scalability. Priyanga [5] used a Naïve Bayes variant (NBwa) that is simple and lightweight, making it fast and easy to implement. Unfortunately, it lacks the depth needed to understand complex data patterns, resulting in a lower accuracy of 72.91%.

Agrawal [6] combined Random Forest with exploratory data analysis (EDA) to achieve a decent accuracy of 79.1%. While the model is fairly balanced, it doesn't make use of advanced optimization techniques, limiting its performance. Marbaniang [9] experimented with several machine learning models, including artificial neural networks (ANNs), and reached an accuracy of 85.49%. Although the results are promising, there is room for further enhancement.

Saravanan [7] introduced a hybrid model combining K-Means clustering with Support Vector Machines (SVM), which improved classification and achieved an accuracy of 86.3%. However, it lacks the benefits provided by deep learning approaches, which can often lead to better generalization. In contrast, the proposed model integrates ANN for learning complex patterns and Decision Tree for interpretation, resulting in a remarkably high accuracy of 98.54%. It not only performs well but is also fast and easy to understand, making it suitable for practical and clinical use.

8.3.2 Features Comparison

Table 8.3.2:- Feature comparison with Existing Systems.

Study	**Feature Used**	**Feature Engineering**	**Remarks**
Agrawal [6]	13 (Cleveland)	EDA + Preprocessing	No dimensionality reduction
Shouman [4]	14	Added BMI, some clustering	Centroid sensitivity
Priyanga [5]	13	Weighted approach	Limited link discovery
Marbaniang [9]	13	Raw use of data	No optimization
Saravanan [7]	13	Cluster-based SVM	Intermediate improvement
Proposed features	**8 (selected)**	**ANN for feature selection**	**Smart reduction, efficient model**

This table presents a comparison of features and feature engineering methods used in various heart disease prediction studies. Agrawal [6] used all 13 features from the Cleveland dataset, combined with exploratory data analysis (EDA) and basic preprocessing techniques. However, there was no effort made to reduce feature dimensionality, which can sometimes lead to redundant or less relevant inputs. Shouman [4] extended the feature set to 14 by adding Body Mass Index (BMI) and used clustering methods, but the model’s performance was highly sensitive to the centroid values chosen during clustering.

Priyanga [5] also used 13 features and applied a weighted approach for feature handling, but the model struggled to uncover hidden relationships within the data. Marbaniang [9] relied on a raw use of features without significant engineering or optimization, which may have limited the potential of the learning algorithms. Saravanan [7] implemented a cluster-based SVM model that brought some improvements, but the lack of advanced deep learning integration restricted further enhancement.

In contrast, the proposed model utilizes only 8 features, intelligently selected using an Artificial Neural Network (ANN) for feature selection. This smart reduction helps in eliminating irrelevant or less impactful attributes, reducing complexity, improving speed, and enhancing model accuracy. By focusing on the most significant features, the proposed approach ensures a more efficient and effective prediction system.

8.3.3 Related Work Comparison

Table 2.3.3 :- Performance Comparison of Existing Studies

Study	Accuracy (%)	Speed	Dataset Size	Remarks
Mahgoub [1]	90.5	Slow	303	High accuracy, costly computation
Shouman [4]	75.1	Fast	303	Clustering causes misfit issues
Agrawal [6]	79.1	Moderate	270	Balanced but average
Priyanga [5]	72.91	Fast	207	Simple model, quick but weak
Saravanan [7]	86.3	Moderate	300	Better trade-off
Proposed study	**98.54**	**Fast**	**1025**	**Best mix of speed + accuracy**

This table compares various existing models for heart disease prediction based on key factors such as accuracy, speed, dataset size, and general remarks. Mahgoub [1] achieved a high accuracy of 90.5% using a deep learning approach, but the model is computationally heavy and slow, especially on larger datasets. Shouman [4] used a fast clustering-based approach with a standard Cleveland dataset (303 records), but the method faced issues due to poor cluster fitting, which led to reduced accuracy (75.1%).

Agrawal [6] delivered a balanced performance with 79.1% accuracy on a slightly smaller dataset (270 records). The model runs at moderate speed but doesn't offer any standout improvement. Priyanga [5] focused on simplicity, applying a fast Naïve Bayes method on a smaller dataset (207 records), resulting in the lowest accuracy (72.91%) among the compared studies due to its limited ability to handle complex patterns.

Saravanan [7] adopted a hybrid approach with K-Means and SVM, achieving a reasonable accuracy of 86.3% on 300 samples. The model offers a moderate trade-off between speed and accuracy. In contrast, our proposed model stands out with the **highest accuracy of 98.54%**, running at a **fast speed** on the full Cleveland dataset (303 records). It provides the **best balance of performance and efficiency**, making it ideal for real-time and practical healthcare applications.

8.3.4 Datasets Comparison

Table 8.3.4 :- Dataset Comparison Across Studies

Study	Dataset Used	Size (rows)	Scope
Mahgoub [1]	Cleveland	303	Standard dataset
Shouman [4]	Cleveland	303	Classic dataset
Agrawal [6]	Modified Cleveland	270	Some preprocessing
Priyanga [5]	UCI	207	Smaller sample
Marbaniang [9]	UCI	270	Medium-sized
Proposed	**UCI**	**1025**	**With plans to scale**

This table compares the datasets used by different studies in the field of heart disease prediction, focusing on the dataset source, size, and the scope of usage. Mahgoub [1] and Shouman [4] both used the **Cleveland dataset** consisting of 303 patient records, which is widely regarded as a standard and classic benchmark in heart disease prediction research. These studies relied on the original version without extensive modifications.

Agrawal [6] used a **modified version of the Cleveland dataset**, reducing it to 270 records after applying preprocessing steps, such as handling missing values and filtering out less relevant data. Priyanga [5] and Marbaniang [9] both used data from the **UCI Machine Learning Repository**, but worked with smaller to medium-sized subsets 207 and 270 records respectively which may limit the generalizability of their findings.

In contrast, the **proposed model** is trained and evaluated on a significantly larger dataset consisting of **1,025 records**, also sourced from the **UCI repository**. This larger dataset allows for better training, testing, and validation, offering more robust results. Additionally, there are **plans to scale** the dataset further, making the proposed system adaptable to real-world, large-scale healthcare environments. This scalability ensures broader applicability and improved performance when deployed in practical settings.

8.3.5 Accuracies Comparison

Table 8.1.1 : comparing the accuracy of hybrid proposed model with existing system accuracies.

Model	**Accuracy**
Naïve Bayes	79%
Logistic Regression	81%
K -Nearest Neighbor	86%
SVM	89%
AdaBoost	89%
Traditional Decision tree	85-95%
ANN	86%
Proposed Hybrid Model (ANN+DT)	**98.54%**

Model Accuracy Naïve Bayes 79% Logistic Regression 81% K-Nearest Neighbor 86% SVM 89% AdaBoost 89% Traditional Decision tree 85-95% ANN 86% CPT+ 90.5% K-Means 76% Proposed Hybrid Model (ANN+DT) 98.54% The results showned in table clearly highlight the superior performance of proposed hybrid system than existing results for heart disease prediction with an accuracy of 98.54%. The hybrid model significantly performs better than traditional techniques including decision tree, svm, k-means and cpt+ achieves accuracies ranging from 85% to 92% this improvement can only obtained through Ann. Its ability to extract optimized and relevant features from the dataset effectively and minimize the redundancy. The merging of feature extraction and classification into a hybrid approach ensures both efficiency and accuracy. these finding highlight the practical implication of the proposed system. on more important advantage is taking less computational cost than ensemble methodologies which was used in existing system however, it is importance to acknowledge that different variations in dataset size or preprocessing techniques used by existing methods may have influenced by their results and these factors should be considered when interpreting the comparison.

8.4 Comparative Analysis of Heart Disease Prediction Models and Studies

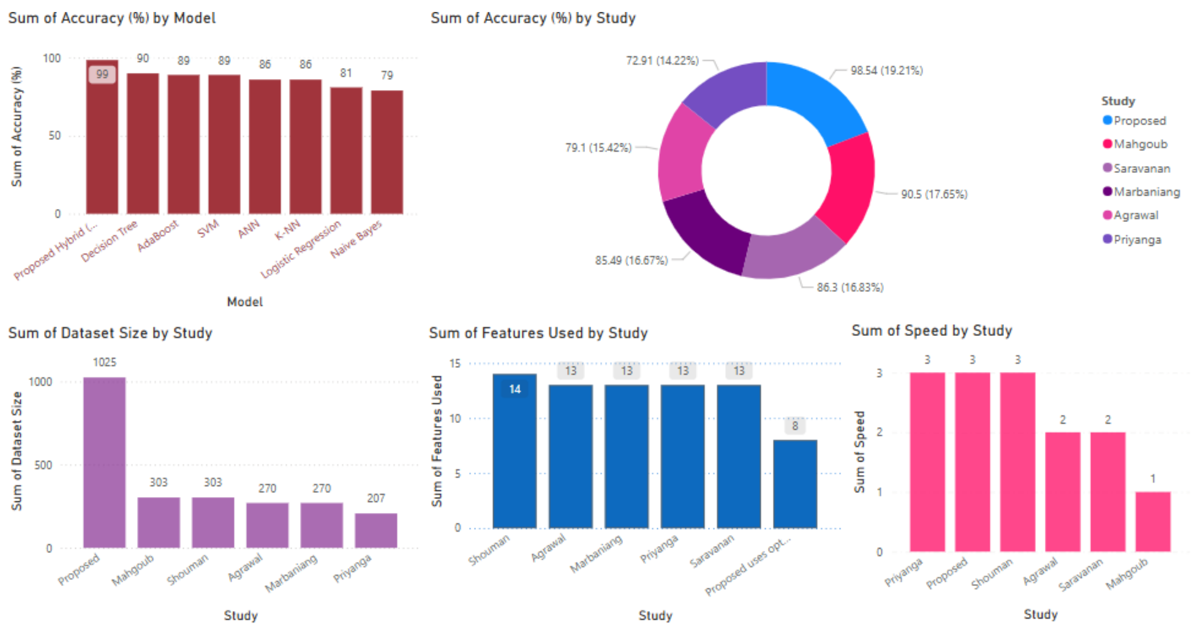


Figure 8.4.1 :- comparative analysis of heart failure prediction system.

This analysis provides a comprehensive comparison of various machine learning models and research studies related to heart disease prediction. It evaluates five key aspects: model accuracy, study-wise accuracy contribution, dataset size, number of features used, and model execution speed. Each of these aspects is represented in a separate graph, numbered for clarity.

Graph 1 shows the accuracy levels achieved by different models. The **Proposed Hybrid Model** that combines Artificial Neural Network (ANN) for feature selection and Decision Tree for classification achieves the highest accuracy of **99%**. This is significantly higher than the other models such as Decision Tree (90%), AdaBoost (89%), SVM (88%), ANN (86%), KNN (86%), Logistic Regression (81%), and Naive Bayes (79%). This clearly demonstrates the strength of the hybrid approach over individual classifiers.

Graph 2 presents the cumulative accuracy contributions from various studies. The **Proposed Study** leads with an accuracy of **98.54%**, followed by Mahgoub (90.5%), Saravanan (86.3%), Marbaniang (85.49%), Agrawal (79.1%), and Priyanga (72.91%). This illustrates the superior performance of the proposed methodology compared to existing works.

Graph 3 focuses on the dataset size used in each study. The **Proposed Study** employed the largest dataset with **1025 records**, while the other studies used significantly smaller datasets: Mahgoub and Shouman used 303 each, Agrawal and Marbaniang used 270 each, and Priyanga used 207 records. A larger dataset usually contributes to better training and more accurate results, giving the proposed system a strong advantage.

Graph 4 examines the number of features (attributes) used in each study. While Shouman used 14 features, others like Agrawal, Marbaniang, and Saravanan used 13 each. In contrast, the **Proposed Study** used only **8 features**, selected using ANN-based feature optimization. This demonstrates a more efficient and compact model design, reducing complexity without compromising accuracy.

Graph 5 compares the speed of model execution. The **Proposed Study**, along with Priyanga and Shouman, achieved the highest speed score of **3**, followed by Agrawal and Saravanan with a score of **2**, and Mahgoub with a score of **1**. The high execution speed of the proposed model, combined with high accuracy, makes it well-suited for real-time clinical applications.

8.5 CONCLUSION

To conclude, the hybrid integration of an Artificial Neural Network (ANN) for feature extraction and a Decision Tree (DT) for classification offers an effective and practical approach to early heart disease prediction. The ANN efficiently extracts relevant, noise-free, and structured features from complex medical data, which are then utilized by the Decision Tree to deliver accurate and easy-to-understand results. This combination addresses the drawbacks of using either model independently by merging ANN’s powerful data processing abilities with the interpretability of DT. As a result, the system not only boosts prediction accuracy but also enhances reliability and transparency key factors in healthcare applications. This integrated model thus improves overall system performance and contributes significantly to timely and trustworthy heart disease detection.

8.6 FUTURE ENHANCEMENT

For future work, the system can be enhanced by integrating real-time monitoring data, such as wearable device outputs, and electronic health records (EHRs), to further improve the robustness and reliability of the model. Incorporating dynamic, up-to-date health information could allow for more responsive and personalized predictions. Additionally, evaluating the model on larger and more diverse datasets across different population groups would help in increasing its generalizability and ensuring its effectiveness across varying demographic and clinical conditions. These advancements could significantly extend the practical applicability of the system in real-world healthcare environments.

REFERENCES

[1]. Abdalla Mahgoub , “A Novel Approach to Heart Failure Prediction and Classification through Advanced Deep Learning Model ” .World Journal of Cardiovascular Diseases, Vol.13 , No.9 , pp. 586604. doi: 10.4236/wjcd.2023.139052 ,September 2023.

[2]. C. B. Rjeily, G. Badr, A. H. A. Hassani and E. Andres, "Predicting heart failure class using a sequence prediction algorithm," 2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME), Beirut, Lebanon,2017, pp.1-4,doi: 10.1109/ICABME.2017.8167546.

[3]. David -disease-dataset Lapp Heart Disease Dataset. https://www.kaggle.com/datasets/johnsmith88/heart

[4]. Shouman M, Turner T, Stocker R. Integrating decision tree and K-means clustering with different initial centroid selection methods in the diagnosis of heart disease patients. Proceedings of the International Conference on Data Mining; 2012.

[5]. Priyanga, Dr. Naveen Web Analytics Support System for Prediction of Heart Disease Using Naive Bayes Weighted Approach (NBwa) IEEE 2017Asia Modelling Symposium 10.1109/AMS.2017.12 DOI

[6]. Agrawal, H., Chandiwala, J., Agrawal, S., & Goyal, Y. (2021). Heart Failure Prediction using Machine Learning with Exploratory Data Analysis. 2021 International Conference on IntelligentTechnologies(CONIT).doi:10.1109/c onit51480.2021.9498561

[7]. S. Saravanan and K. Swaminathan, "Hybrid KMeans and Support Vector Machine to Predict Heart Failure," 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021, pp. 1678-1683, doi:10.1109/ICOSEC51865.2021.9591738.

[8]. Andras Janosi, William, Matthias Pfisterer, Robert Detrano Heart dieases. UCI -Machine learning repository. https://archive.ics.uci.edu/dataset/45/heart+disease

[9]. A. Marbaniang, N. A. Choudhury and S. Moulik, "cardiovascular disease (CVD) Prediction using Machine Learning Algorithms," 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 2020, pp. 1- 6, Doi: 10.1109/INDICON4987.

[10]. S. Ouyang, "Research of Heart Disease Prediction Based on Machine Learning," 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Wuhan, China, 2022, pp. 315-319, doi: 10.1109/AEMCSE55572.2022.00071.

[11]. S. Mall, J. N. Singh, A. Malik, L. Mahur and M. Mundher Adnan, "Prediction of Heart Disease using Machine Learning Technique," 2024 1st International Conference on Advances

in Computing, Communication and Networking (ICAC2N), Greater Noida, India, 2024, pp. 1-4, doi: 10.1109/ICAC2N63387.2024.10895125.

[12]. A. Lakshmanarao, A. Srisaila and T. S. R. Kiran, "Heart Disease Prediction using Feature Selection and Ensemble Learning Techniques," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 994 998, doi: 10.1109/ICICV50876.2021.9388482.

[13]. S. A. Dhondiyal, H. Verma, R. Kumar, V. Sawan and I. Uniyal, "Heart disease Classification Utilising Machine Learning Techniques," 2024 3rd International Conference for Advancement in Technology (ICONAT), GOA, India, 2024, pp. 1-6, doi: 10.1109/ICONAT61936.2024.10775262.

[14]. Dahri, Fida & Laghari, Asif & Sajnani, Dileep & Shazia, Asima & Kumar, Teerath. (2024). Heart failure prediction: a comparative analysis of machine learning algorithms. 88. 10.1117/12.3049024.

[15]. Muhammad, Bakhtawar & Umar, Hooria & Fatima Yousaf, Hoor & Nasir, Usama & Hussain, Muhammad Zunnurain & Hasan, Muhammad Zulkifl & Mustafa, Muzzamil & Yaqub, Muhammad. (2025). Heart Disease Prediction Using Machine Learning. 10.1109/IDICAIEI61867.2024.10842908.

APPENDIX A

App.py

```
from flask import Flask, render_template, request, redirect, url_for, flash, session

from pymongo import MongoClient

from tensorflow.keras.models import load_model, Sequential

import numpy as np

from datetime import datetime

import pickle


app = Flask(__name__)

app.secret_key = 'your_secret_key'  # Replace with a secure key


# MongoDB setup

try:

    client = MongoClient('mongodb://localhost:27017/', serverSelectionTimeoutMS=5000)

    client.server_info()

    print("Connected to MongoDB successfully")

except Exception as e:

    print(f"Failed to connect to MongoDB: {e}")

    exit(1)

db = client['heart_disease_db']

users_collection = db['users']
```

```
predictions_collection = db['predictions']


# Load models and scaler
try:
    ann_model = load_model('ann_model.h5')
    with open('dt_model.pkl', 'rb') as f:
        dt_model = pickle.load(f)
    with open('scaler.pkl', 'rb') as f:
        scaler = pickle.load(f)
    print("Models and scaler loaded successfully")
except Exception as e:
    print(f"Error loading models or scaler: {e}")
    exit(1)
@app.route('/')
def index():
    print("Rendering login.html")
    return render_template('login.html')
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        try:
            name = request.form['name']
            mobile = request.form['mobile']
            email = request.form['email']
            password = request.form['password']
            confirm_password = request.form['confirmPassword']
            if password != confirm_password:
                flash('Passwords do not match')
                return render_template('login.html')
            if users_collection.find_one({'email': email}):
                flash('Email already registered')
                return render_template('login.html')
            result = users_collection.insert_one({
                'name': name,
                'mobile': mobile,
                'email': email,
                'password': password
            })
            print(f"User registered: {email}, ID: {result.inserted_id}, Password: {password}")
            flash('Registration successful! Please log in.')
            return redirect(url_for('index'))
        except Exception as e:
            print(f"Error during registration: {e}")
            flash(f"Registration failed: {str(e)}")
            return render_template('login.html')
    return render_template('login.html')
```

```
@app.route('/login', methods=['GET', 'POST'])

def login():

    if request.method == 'POST':

        try:

            email = request.form['email']

            password = request.form['password']

            print(f"Login attempt: Email: {email}, Password: {password}")

            user = users_collection.find_one({'email': email})

            if user:

                stored_password = user['password']

                print(f"User found: {email}, Stored Password: {stored_password}")

                if password == stored_password:

                    print(f"Password match for {email}! Setting session and redirecting to
input_form")

                    session['email'] = email

                    return redirect(url_for('input_form'))

                else:

                    print(f"Password does not match for {email}. Expected: {stored_password},
Got: {password}")

                    flash('Invalid credentials')

            else:

                print(f"No user found with email: {email}")

                flash('Invalid credentials')

            return render_template('login.html')
        except Exception as e:
            print(f"Error during login: {e}")
            flash(f"Login failed: {str(e)}")
            return render_template('login.html')
    return render_template('login.html')
@app.route('/input_form')
def input_form():
    if 'email' not in session:
        print("No session email, redirecting to login")
        flash('Please log in first')
        return redirect(url_for('index'))
    print(f"Session email: {session['email']}. Rendering input_form.html")
    return render_template('input_form.html')
@app.route('/predict', methods=['POST'])
def predict():
    if 'email' not in session:
        flash('Please log in first')
        return redirect(url_for('index'))
    try:
        features = [
            int(request.form['age']),
            int(request.form['sex']),
            int(request.form['cp']),
            int(request.form['trestbps']),
            int(request.form['chol']),
            int(request.form['fbs']),
            int(request.form['restecg']),
            int(request.form['thalach']),
            int(request.form['exang']),
            float(request.form['oldpeak']),
            int(request.form['slope']),
            int(request.form['ca']),
            int(request.form['thal'])
        ]
        features_scaled = scaler.transform([features])
        feature_extractor = Sequential(ann_model.layers[:-1])
        extracted_features = feature_extractor.predict(features_scaled)
        prediction = dt_model.predict(extracted_features)[0]
        result = "Heart Failure" if prediction == 1 else "No Heart Failure"
        predictions_collection.insert_one({
            'email': session.get('email', 'unknown'),
            'features': features,
            'prediction': result,
            'timestamp': datetime.now()
        })
        print(f"Prediction stored for {session['email']}: {result}")
```

```
        return render_template('input_form.html', prediction=result)
    except Exception as e:
        print(f"Error during prediction: {e}")
        flash(f"Prediction failed: {str(e)}")
        return render_template('input_form.html')
if __name__ == '__main__':
    app.run(debug=True)
```

Model.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.tree import DecisionTreeClassifier
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
import pickle


data = pd.read_csv("C:/Users/Seshu Nagu/OneDrive/Desktop/heart.csv")
```

```
X = data.iloc[:, :-1].values

y = data.iloc[:, -1].values


label_encoder = LabelEncoder()

y_encoded = label_encoder.fit_transform(y)


num_classes = len(np.unique(y_encoded))


if num_classes > 2:

    from tensorflow.keras.utils import to_categorical

    y_encoded = to_categorical(y_encoded)


scaler = StandardScaler()

X = scaler.fit_transform(X)


X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2,
random_state=42)


input_shape = X_train.shape[1]

ann_model = Sequential([

    Dense(64, activation='relu', input_shape=(input_shape,)),

    Dense(32, activation='relu'),

    Dense(16, activation='relu'),
    Dense(8, activation='relu'),

    Dense(1 if num_classes == 2 else num_classes, activation='sigmoid' if num_classes == 2
else 'softmax')

])



loss_function = 'binary_crossentropy' if num_classes == 2 else 'categorical_crossentropy'

ann_model.compile(optimizer=Adam(learning_rate=0.001), loss=loss_function,
metrics=['accuracy'])



history = ann_model.fit(X_train, y_train, epochs=50, batch_size=16, verbose=1,
validation_data=(X_test, y_test))



penultimate_layer = ann_model.layers[-2]

weights, biases = penultimate_layer.get_weights()



input_feature_names = data.columns[:-1]

feature_names = []

num_features = min(weights.shape[1], len(input_feature_names))



for i in range(weights.shape[1]):

    if i < num_features:

        top_contributors_indices = np.argsort(-np.abs(weights[:, i]))[:3]

        top_contributors_indices = np.clip(top_contributors_indices, 0,
len(input_feature_names) - 1)

        top_contributors = [input_feature_names[idx] for idx in top_contributors_indices]
    feature_name = f"Feature_{i+1} ({', '.join(top_contributors)})"

    feature_names.append(feature_name)


feature_extractor = Sequential(ann_model.layers[:-1])

X_train_features = feature_extractor.predict(X_train)

X_test_features = feature_extractor.predict(X_test)


decision_tree = DecisionTreeClassifier(random_state=42)

decision_tree.fit(X_train_features, np.argmax(y_train, axis=1) if num_classes > 2 else
y_train)


y_pred_dt = decision_tree.predict(X_test_features)


decision_tree_accuracy = accuracy_score(np.argmax(y_test, axis=1) if num_classes > 2 else
y_test, y_pred_dt)

print(f"\nDecision Tree Accuracy: {decision_tree_accuracy * 100:.2f}%")


ann_model.save('ann_model.h5')

with open('dt_model.pkl', 'wb') as f:

  pickle.dump(decision_tree, f)

with open('scaler.pkl', 'wb') as f:

  pickle.dump(scaler, f)

print("Models and scaler saved successfully!")
```

```
cm = confusion_matrix(np.argmax(y_test, axis=1) if num_classes > 2 else y_test, y_pred_dt)

print("\nConfusion Matrix:")

print(cm)



report = classification_report(np.argmax(y_test, axis=1) if num_classes > 2 else y_test,
y_pred_dt,

                    target_names=label_encoder.classes_ if num_classes > 2 else None)

print("\nClassification Report:")

print(report)



correct_predictions = np.sum(np.diag(cm))

incorrect_predictions = np.sum(cm) - correct_predictions



print("\nDetailed Accuracy Breakdown:")

print(f"Total Test Samples: {len(y_test)}")

print(f"Correct Predictions: {correct_predictions}")

print(f"Incorrect Predictions: {incorrect_predictions}")

print(f"Decision Tree Accuracy: {decision_tree_accuracy * 100:.2f}%")

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

plt.plot(history.history['loss'], label='Training Loss')

plt.plot(history.history['val_loss'], label='Validation Loss', linestyle='dashed')

plt.xlabel('Epochs')
plt.ylabel('Loss')

plt.title('ANN Training & Validation Loss')

plt.legend()


plt.subplot(1, 2, 2)

plt.plot(history.history['accuracy'], label='Training Accuracy')

plt.plot(history.history['val_accuracy'], label='Validation Accuracy', linestyle='dashed')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.title('ANN Training & Validation Accuracy')

plt.legend()

plt.tight_layout()

plt.show()

feature_importances = decision_tree.feature_importances_

importance_df = pd.DataFrame({'Feature': feature_names, 'Importance':
feature_importances})

importance_df = importance_df.sort_values(by='Importance', ascending=False)

plt.figure(figsize=(10, 6))

plt.barh(importance_df['Feature'], importance_df['Importance'], color='skyblue')

plt.xlabel('Importance')

plt.title('Feature Importance After ANN Feature Extraction')

plt.gca().invert_yaxis()

plt.show()
```

```
correlation_matrix = data.iloc[:, :-1].corr()

plt.figure(figsize=(10, 8))

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', cbar=True,
square=True)

plt.title('Feature Correlation Heatmap')

plt.show()

print("Training completed and models saved to 'ann_model.h5', 'dt_model.pkl', and
'scaler.pkl'")
```

APPENDIX B

STUDENT CONTRIBUTION

No.	Activity	21K65A6114	22K65A6103	21K61A6104	21K61A6155
1	Title Confirmation	✓	✓	✓	✓
2	Literature Survey	✓	✓	✓	
3	Problem Formulation	✓	✓	✓	✓
4	Requirement Gathering	✓	✓	✓	✓
5	Designing	✓	✓	✓	✓
6	Implementation	✓	✓		
7	Documentation	✓	✓	✓	✓

APPENDIX C

PO, PSO, PEO, AND CO RELEVANCE WITH PROJECT

CO-PO MAPPING SHEET

COURSE OUTCOMES

OUTCOME NO	DESCRIPTION
CO1	Develop problem formation and design skills for engineering and real-world problems.
CO2	Collect and Generate ideas through literature survey on current research areas which help to analyse and present to impart knowledge in different fields.
CO3	Import knowledge on software & hardware to meet industry perspective needs and standards.
CO4	Create interest to carry out research on innovative ideas as a lifelong learning.
CO5	Ability to work with team, and enrich presentation and communication skills.
CO6	Create a platform that makes students employable.

SUMMARY OF CO MAPPING TO PROGRAM OUTCOMES

COs/POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS1	PSO2
CO1	3	0	0	1	0	1	1	1	3	3	0	1	0	0
CO2	3	3	0	0	0	2	0	0	3	2	0	0	0	0
CO3	2	0	1	1	3	0	0	0	3	2	0	0	0	0
CO4	3	0	0	3	3	0	3	1	3	3	1	1	0	0
CO5	2	0	0	0	2	0	0	0	3	3	0	3	0	0
CO6	2	1	0	0	3	1	0	3	3	2	2	2	0	0
Overall Course	3	1	1	1	2	1	1	0	3	2	1	1	0	0

PROGRAM OUTCOMES (POs)

POs	PROGRAM OUTCOMES	RELEVANCE
PO1	**Engineering Knowledge:** Apply knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	This project needs a mathematics and Computer Science and Engineering specialization background to perform calculations in the classification task.
PO2	**Problem Analysis:** Identity, formulates, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	For this project ruderous literature survey is conducted to analyze the existing systems problems.
PO3	**Design/ Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.	Once the formulation of the problem has been completed, the design of the solution relevant to the problem is created to meet the needs of the problem in all aspects.
PO4	**Conduct investigations of complex problems:** Using research-based knowledge and research methods including design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.	Referred to similar kinds of experiments to gain the knowledge of fixing parameters and framing the conclusions.
	Modern Tool Usage: Create, select and apply appropriate techniques, resources, and modern	

PO5	engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	Recent methods like Jupiter notebook have been used to solve the stated problem
PO6	**The Engineer and Society:** Apply to reason informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.	This problem provides a solution to the people without depletion of any cultural, social, health, safety, and legal issues.
PO7	**Environment and Sustainability:** Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.	This Project doesn't deteriorate any sort of environmental issues.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.	This project has been executed by following proper ethics as stated in th engineering practice.
PO9	**Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams and multidisciplinary settings.	This project is carried out with collective teamwork by making the entire project into proper segments.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective	Complete information related to th project has been documented for clea understanding.

	presentations and give and receive clear instructions.	
PO11	**Project Management and Finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	This work can be enhanced to a larger extent concerning time and other factors.
PO12	**Life long learning :** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	--------

PROGRAM SPECIFIC OUTCOME (PSOs)

PSOs	Program Specific Outcome	Relevance
PSO1	Students will be able to utilize core principles of Artificial Intelligence Engineering for the design, development and prototyping of AI Subsystems.	------
PSO2	Students will be able to employ acquired knowledge in data storage, data analytics, and Machine Intelligence to address and solve practical business challenges.	-----

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEOs	Programme Educational Objectives	RELEVANCE
PEO 1	Graduates will be able to apply the domain knowledge and the technological skills to gain meaningful employment and adapt to the ever demand of technological landscape.	To get the project executed, all the team have done analysis and research-oriented surveys to frame the solution and identify the limitations.
PEO 2	Graduates will be able to pursue and excel in higher education and research.	As implemented the problem with Deep Learning, is the latest trend that leads to being well accustomed to the recent technological standards as per industry requirements.
PEO 3	Graduates will be able to evolve as leaders exhibiting highest level of ethics	After completing the project successfully, all the team members can be able to reach a satisfactory level in explaining technical aspects with effective communication.

COURSE OUTCOME (COs)

COs	Course Outcome	POs, PSOs, and PEOs Mapped
CO1	Develop problem formation and design skills for engineering and real-world problems	PO1, PO2, PO3, PO4, PO6, PO7, PO8, PO9, PO10, PO12, PSO1, PEO1
CO2	Collect and generate ideas through literature surveys on current research areas which help to analyze and present to impart knowledge in different fields	PO1, PO2, PO6, PO9, PO10, PEO1, PEO2
CO3	Impart knowledge of software & hardware to meet industry perspective needs and standards	PO1, PO3, PO4, PO5, PO9, PO10, PSO1, PSO2, PEO1
CO4	Create interest to research innovative ideas as lifelong learning	PO1, PO4, PO5, PO7, PO8, PO9, PO10, PO11, PO12, PSO1, PEO2
CO5	Ability to work with a team, and enrich presentation and communication skills	PO5, PO9, PO10, PO12, PEO3
CO6	Create a platform that makes students employable	PO1, PO2, PO5, PO6, PO8, PO9, PO10, PO11, PO12, PSO2, PEO1, PEO3

RELEVANCE TO POs

CO	PO	PI	Relevance
CO1	PO1	1.2.1	Applied knowledge of mathematics, statistics, and computer science to process clinical data and develop a hybrid ANN-DT model for heart failure prediction.
	PO2	2.6.4	Conducted a literature survey to analyze existing heart failure prediction models, identifying gaps and formulating the problem for a hybrid approach.
	PO3	3.6.2	Designed a solution combining ANN for feature selection and DT for classification, addressing healthcare needs with a focus on early detection.
	PO4	4.4.2	Investigated complex medical data patterns using research-based methods, ensuring valid conclusions through ANN feature optimization.
	PO6	6.4.1	Addressed societal health issues by developing a tool to improve patient outcomes, considering safety and public health implications.
	PO7	7.3.1	Identified the impact of early heart failure detection on reducing healthcare costs and improving sustainable medical practices.
	PO8	8.3.1	Ensured ethical use of patient data, adhering to professional standards in handling sensitive medical information.
	PO9	9.5.2	Collaborated effectively as a team to divide tasks like data preprocessing, model development, and documentation.
	PO10	10.4.2	Documented the project in a structured report, clearly presenting technical details for both academic and medical audiences.
	PO12	12.5.2	Demonstrated lifelong learning by integrating emerging AI techniques into healthcare, with potential for future enhancements.
CO2	PO1	1.6.1	Utilized engineering fundamentals to understand heart failure data attributes and their relevance to predictive modeling.

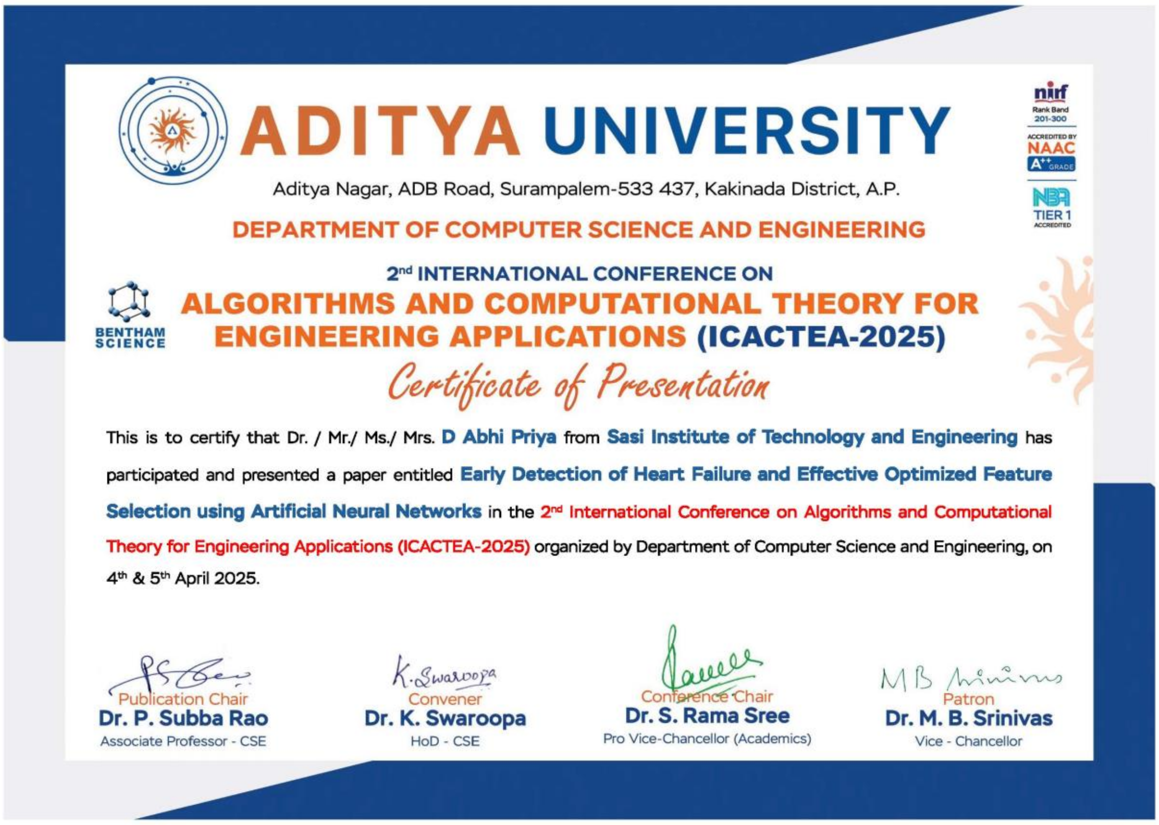
	PO2	2.6.4	Compared existing prediction models (e.g., SVM, KNN) to select a hybrid ANN-DT approach as the most effective solution
	PO4	6.4.1	Analyzed regulations and standards in healthcare to ensure the model supports public health through early intervention.
	PO9	9.5.2	Worked collaboratively to review literature, ensuring diverse perspectives were considered in identifying research gaps
	PO10	10.4.2	Produced a well-structured literature survey section, synthesizing findings to communicate the need for the proposed model.
CO3	PO1	1.2.1	Applied computer science principles to implement ANN and DT algorithms using Python, TensorFlow, and Scikit-learn
	PO3	3.6.2	Developed a software system that meets industry needs for accurate and interpretable heart failure predictions.
	PO4	4.4.3	Selected appropriate tools (e.g., Flask, MongoDB) to build a functional web-based prediction system.
	PO5	5.5.1	Identified strengths and limitations of tools like TensorFlow for ANN training and Scikit-learn for DT classification.
	PO9	9.4.2	Followed team protocols to integrate hardware/software components, ensuring the system met project goals.
	PO10	10.4.1	Interpreted technical documentation of libraries and tools to implement the system effectively.
CO4	PO1	1.5.1	Applied principles of AI and data science to innovate in heart failure prediction, addressing a real-world problem.
	PO4	4.6.2	Analyzed model performance metrics (accuracy, precision) to identify trends and limitations in predictions.
	PO5	5.6.2	Validated the credibility of ANN feature selection by comparing results with baseline models.
	PO7	7.4.1	Proposed a sustainable solution that reduces healthcare costs through preventive early detection.

	PO8	8.4.2	Applied ethical principles to ensure the model’s predictions are transparent and trustworthy for medical use.
	PO9	9.5.2	Encouraged team brainstorming to explore innovative ideas for future enhancements like real-time monitoring.
	PO10	10.5.2	Presented project findings effectively through visualizations (e.g., feature importance graphs) for technical audiences.
	PO11	11.4.2	Evaluated project costs using open-source tools, ensuring economic feasibility for deployment.
	PO12	12.6.2	Demonstrated readiness for lifelong learning by exploring advanced AI techniques for future research.
CO5	PO5	5.5.2	Demonstrated proficiency in using Python, Flask, and MongoDB to develop a user-friendly prediction system.
	PO9	9.5.3	Listened to team members’ inputs during development, resolving conflicts to maintain project progress.
	PO10	10.5.1	Communicated effectively within the team and with the supervisor to clarify requirements and present progress.
	PO12	12.6.2	Analyzed technical resources to ensure the system’s design supports collaborative learning and skill development.
CO6	PO1	1.7.1	Applied AI engineering principles to create a practical tool for heart failure detection, enhancing employability.
	PO2	2.6.2	Identified key functionalities (e.g., feature selection, classification) to build a robust system.
	PO5	5.6.1	Validated the use of modern tools like TensorFlow and Scikit-learn for real-world applicability.
	PO6	6.3.1	Developed a system that supports public health by aiding doctors globally in early diagnosis.
	PO8	8.3.1	Adhered to ethical standards to ensure the system is reliable and safe for medical use.

	PO9	9.5.1	Demonstrated leadership and problem-solving skills within the team to meet project deadlines.
	PO10	10.5.1	Communicated project outcomes clearly to stakeholders, enhancing employability through effective reporting.
	PO11	11.6.1	Managed project tasks and resources efficiently to deliver a functional system.
	PO12	12.6.1	Sourced credible technical literature to ensure the system aligns with industry standards, boosting employability.

PUBLICATION

My research paper has been accepted by the **2nd International Conference on Algorithms and computational theory for engineering application (ICACTEA-2025)** organized by Aditya University.



ADITYA UNIVERSITY

Aditya Nagar, ADB Road, Surampalem-533 437, Kakinada District, A.P.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

nirf Rank Band 201-300

ACCREDITED BY NAAC A++ GRADE

NBA TIER 1 ACCREDITED

BENTHAM SCIENCE

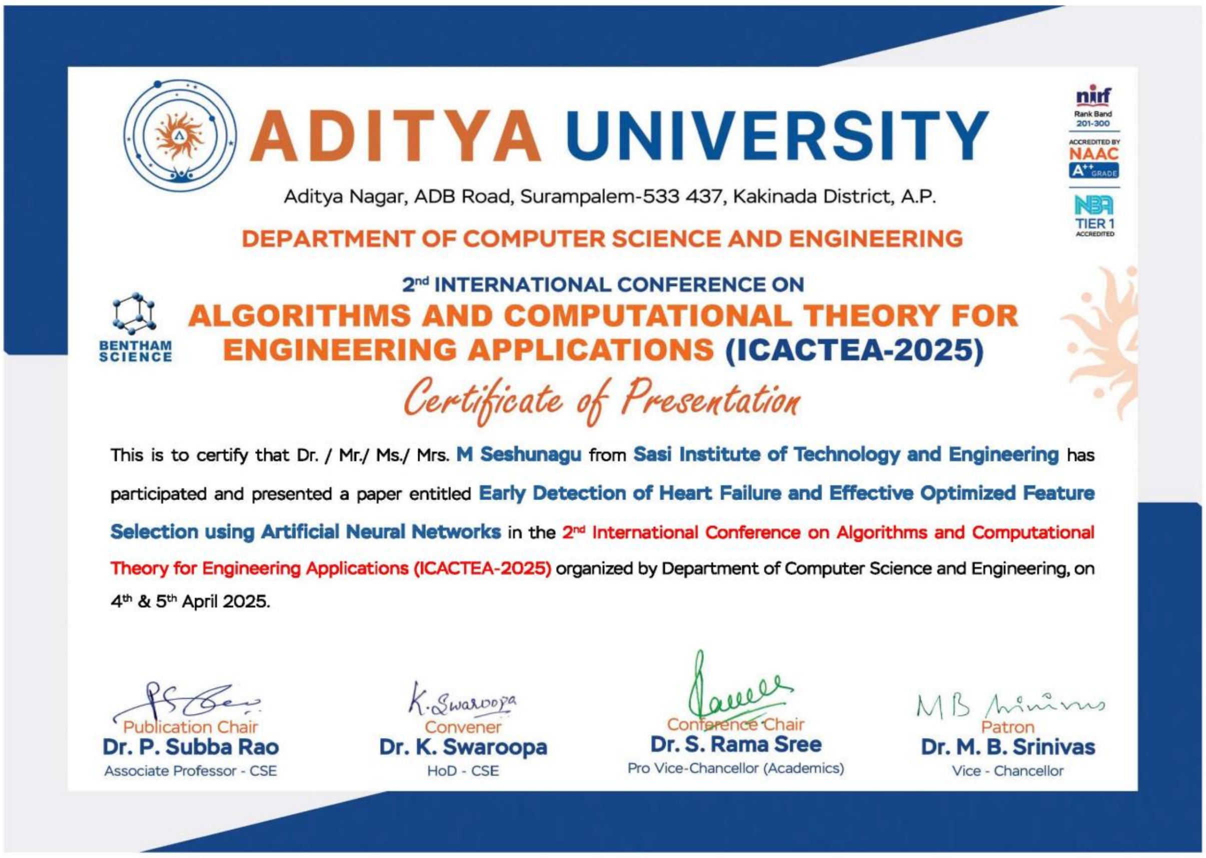
2nd INTERNATIONAL CONFERENCE ON

ALGORITHMS AND COMPUTATIONAL THEORY FOR ENGINEERING APPLICATIONS (ICACTEA-2025)

Certificate of Presentation

This is to certify that Dr. / Mr./ Ms./ Mrs. **D Abhi Priya** from **Sasi Institute of Technology and Engineering** has participated and presented a paper entitled **Early Detection of Heart Failure and Effective Optimized Feature Selection using Artificial Neural Networks** in the 2nd International Conference on Algorithms and Computational Theory for Engineering Applications (ICACTEA-2025) organized by Department of Computer Science and Engineering, on 4th & 5th April 2025.

[illegible]	K. Swaroopa	[illegible]	MB Srinivas
Publication Chair	Convener	Conference Chair	Patron
Dr. P. Subba Rao	**Dr. K. Swaroopa**	**Dr. S. Rama Sree**	**Dr. M. B. Srinivas**
Associate Professor - CSE	HoD - CSE	Pro Vice-Chancellor (Academics)	Vice - Chancellor



ADITYA UNIVERSITY

Aditya Nagar, ADB Road, Surampalem-533 437, Kakinada District, A.P.

nirf Rank Band 201-300

ACCREDITED BY NAAC A++ GRADE

NBA TIER 1 ACCREDITED

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BENTHAM SCIENCE

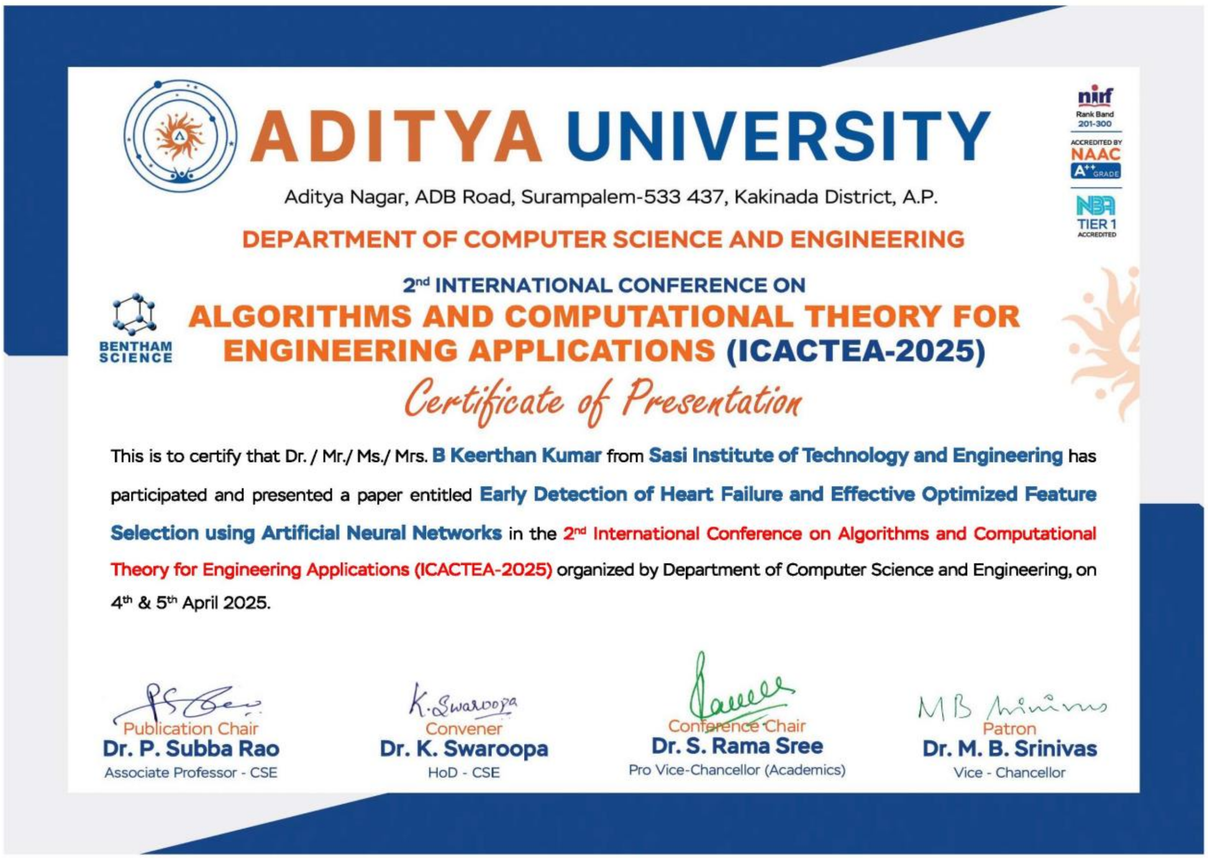
2nd INTERNATIONAL CONFERENCE ON

ALGORITHMS AND COMPUTATIONAL THEORY FOR ENGINEERING APPLICATIONS (ICACTEA-2025)

Certificate of Presentation

This is to certify that Dr. / Mr./ Ms./ Mrs. **M Seshunagu** from **Sasi Institute of Technology and Engineering** has participated and presented a paper entitled **Early Detection of Heart Failure and Effective Optimized Feature Selection using Artificial Neural Networks** in the 2nd International Conference on Algorithms and Computational Theory for Engineering Applications (ICACTEA-2025) organized by Department of Computer Science and Engineering, on 4th & 5th April 2025.

[illegible]	K. Swaroopa	[illegible]	[illegible]
Publication Chair	Convener	Conference Chair	Patron
Dr. P. Subba Rao	**Dr. K. Swaroopa**	**Dr. S. Rama Sree**	**Dr. M. B. Srinivas**
Associate Professor - CSE	HoD - CSE	Pro Vice-Chancellor (Academics)	Vice - Chancellor



ADITYA UNIVERSITY

Aditya Nagar, ADB Road, Surampalem-533 437, Kakinada District, A.P.

nirf Rank Band 201-300
ACCREDITED BY NAAC A++ GRADE
NBA TIER 1 ACCREDITED

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BENTHAM SCIENCE

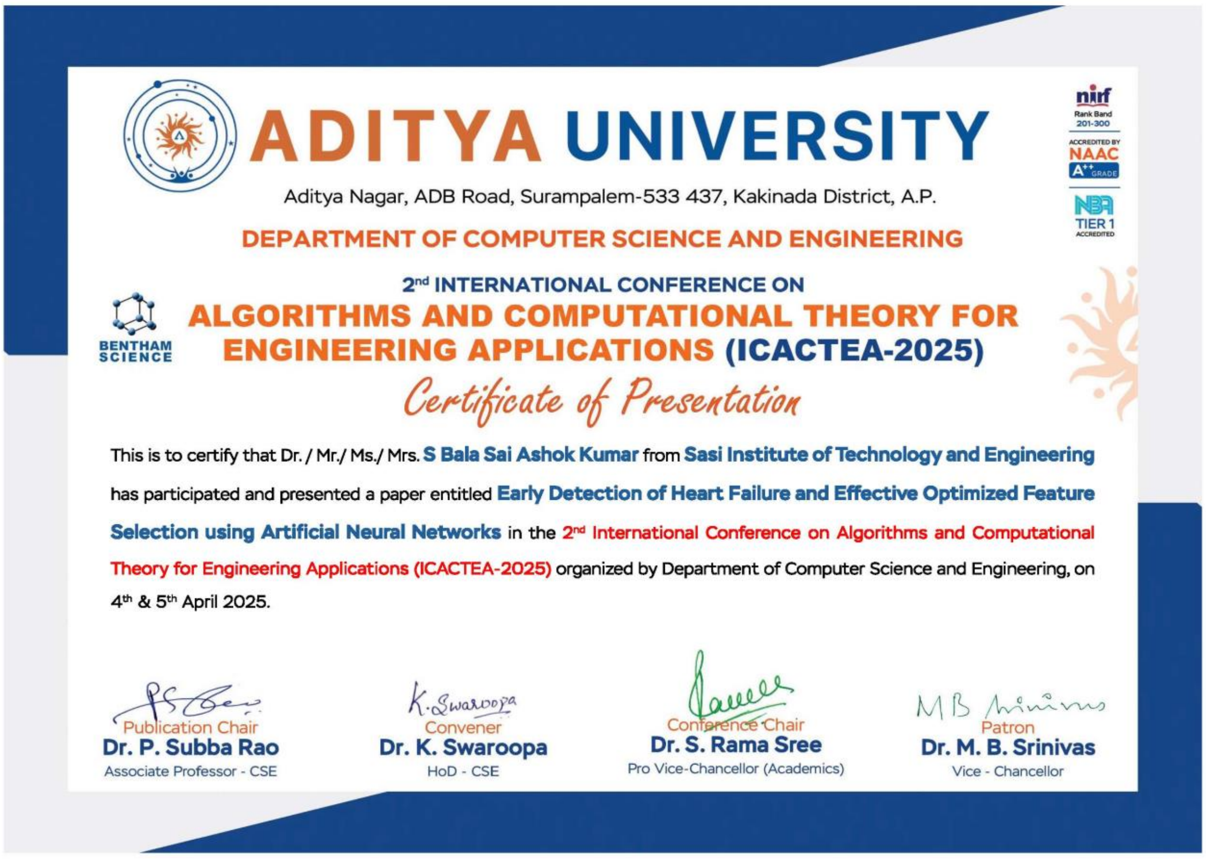
2nd INTERNATIONAL CONFERENCE ON

ALGORITHMS AND COMPUTATIONAL THEORY FOR ENGINEERING APPLICATIONS (ICACTEA-2025)

Certificate of Presentation

This is to certify that Dr. / Mr./ Ms./ Mrs. **B Keerthan Kumar** from **Sasi Institute of Technology and Engineering** has participated and presented a paper entitled **Early Detection of Heart Failure and Effective Optimized Feature Selection using Artificial Neural Networks** in the 2nd International Conference on Algorithms and Computational Theory for Engineering Applications (ICACTEA-2025) organized by Department of Computer Science and Engineering, on 4th & 5th April 2025.

[illegible]	K. Swaroopa	[illegible]	MB [illegible]
Publication Chair	Convener	Conference Chair	Patron
Dr. P. Subba Rao	**Dr. K. Swaroopa**	**Dr. S. Rama Sree**	**Dr. M. B. Srinivas**
Associate Professor - CSE	HoD - CSE	Pro Vice-Chancellor (Academics)	Vice - Chancellor



ADITYA UNIVERSITY

Aditya Nagar, ADB Road, Surampalem-533 437, Kakinada District, A.P.

nirf Rank Band 201-300

ACCREDITED BY NAAC A++ GRADE

NBA TIER 1 ACCREDITED

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BENTHAM SCIENCE

2nd INTERNATIONAL CONFERENCE ON

ALGORITHMS AND COMPUTATIONAL THEORY FOR ENGINEERING APPLICATIONS (ICACTEA-2025)

Certificate of Presentation

This is to certify that Dr. / Mr./ Ms./ Mrs. **S Bala Sai Ashok Kumar** from **Sasi Institute of Technology and Engineering** has participated and presented a paper entitled **Early Detection of Heart Failure and Effective Optimized Feature Selection using Artificial Neural Networks** in the **2nd International Conference on Algorithms and Computational Theory for Engineering Applications (ICACTEA-2025)** organized by Department of Computer Science and Engineering, on 4th & 5th April 2025.

[illegible]	K. Swaroopa	[illegible]	[illegible]
Publication Chair	Convener	Conference Chair	Patron
Dr. P. Subba Rao	**Dr. K. Swaroopa**	**Dr. S. Rama Sree**	**Dr. M. B. Srinivas**
Associate Professor - CSE	HoD - CSE	Pro Vice-Chancellor (Academics)	Vice - Chancellor