UNIVERSITI TEKNOLOGI MARA

DIAGNOSIS AND TREATMENT RECOMMENDER SYSTEM FOR MYOCARDIAL INFARCTION USING DECISION TREE AND SUPPORT VECTOR MACHINES (SVM)

WAN MARZUQI AMRIN BIN WAN MANSOR

BACHELOR OF COMPUTER SCIENCE (Hons)

January 2025

Universiti Teknologi MARA

Diagnosis and Treatment Recommender System for Myocardial Infarction using Decision Tree and Support Vector Machines (SVM)

Wan Marzuqi Amrin Bin Wan Mansor

Thesis submitted in fulfilment of the requirement for Bachelor of Computer Science (Hons.) College of Computing, Informatic and Mathematic

January 2025

SUPERVISOR APPROVAL

DIAGNOSIS AND TREATMENT RECOMMENDER SYSTEM FOR MYOCARDIAL INFARCTION USING DECISION TREE AND SUPPORT VECTOR MACHINES (SVM)

By

WAN MARZUQI AMRIN BIN WAN MANSOR
2023125489

This thesis was prepared under the supervision of the project supervisor, Zawawi bin Ismail @ Abdul Wahab. It was submitted to the College of Computing, Informatic and Mathematic and was accepted in partial fulfilment of the requirements for the degree of Bachelor of Computer Science (Hons.)

Approved by

……………………………

Zawawi bin Ismail @ Abdul Wahab
Project Supervisor

JANUARY 25, 2025

STUDENT DECLARATION

I certify that this thesis and the project to which it refers is the product of my own work and that any idea or quotation from the work of other people, published or otherwise are fully acknowledged in accordance with the standard referring practices of the discipline.

……………………….

WAN MARZUQI AMRIN BIN WAN MANSOR

2023125489

JANUARY 25, 2025

ACKNOWLEDGEMENT

Alhamdulillah, praises and thanks to Allah because of His Almighty and His utmost blessings, I was able to finish this research within the time duration given. Firstly, my special thanks go to my supervisor, Sir Zawawi bin Ismail @ Abdul Wahab for his invaluable guidance, patience and support throughout this research project. His expertise and insightful feedback were instrumental in shaping this work.

Special appreciation also goes to my beloved parents for their endless love, prayers and encouragement. Their unwavering support has been my pillar of strength and motivation during this challenging journey.

I would like to give my gratitude to my dearest friends for always being there for me. Their companionship, understanding and moral support were invaluable especially during the difficult times of this project.

Last but not least, I want to thank me for believing in me. I want to thank me for doing all this hard work. I want to thank me for having no days off. I want to thank me for never quitting. I want to thank me for always being a giver and trying to give more than I receive. I want to thank me for trying to do more right than wrong. I want to thank me for just being me at all times.

To all those who have contributed directly or indirectly to the successful completion of this research, I am truly grateful. Thank you.

ABSTRACT

Myocardial infarction which commonly known as heart attack is a critical medical condition that demands accurate diagnosis followed by an appropriate treatment plan. This project presents the development process of the prototype for diagnosis and treatment recommender system for myocardial infarction using decision tree and support vector machine (SVM) algorithms. Healthcare professionals can benefit from this prototype system that uses ECG images for myocardial infarction diagnosis while recommending proper treatments based on patient clinical information. The prototype functions by initially allowing the user to upload an ECG image which will be processed using SVM for feature extraction and classification. If the ECG image is classified as indicative of myocardial infarction, the user inputs additional patient clinical data. The decision tree algorithm functions after this point. The prototype processes collected clinical data using these algorithms to confirm diagnoses while determining the level of patient severity. The user interface of the prototype is designed to be user-friendly, minimizing the risk of user error and ensuring smooth workflow. The ultimate goal of this system is to improve patient outcomes by enabling precise diagnosis and personalized treatment recommendations. The support vector machines (SVM) model achieved an accuracy of 94.48% while the decision tree model achieved an accuracy of 96.47%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SVM	Support Vector Machines
ECG	Electrocardiogram

CHAPTER 1

INTRODUCTION

This chapter provides the background and rationale for the study. It also gives the details of the significance of diagnosis and treatment recommender system for myocardial infarction using decision tree and support vector machines (SVM), the issues and problems that led to this research.

1.1 Background of Study

Myocardial infarction, commonly known as heart attack, is a severe medical condition resulting from the interruption of blood flow to the heart, often leading to tissue damage or death. According to the World Health Organization, cardiovascular diseases (CVDs) are the leading cause of death globally with an estimated 17.9 million people dying from CVDs in 2019, representing 32% of all global deaths (Cardiovascular diseases (CVDs), 2021).
Current diagnostic and treatment for myocardial infarction involve a multifaceted approach aimed at identifying and addressing the condition's underlying causes. The diagnostic methods begin with initial and serial electrocardiography (ECG) and serial measurements of cardiac biomarkers to help distinguish between unstable angina, ST-segment elevation myocardial infarction (STEMI) and non-ST-segment myocardial infarction (NSTEMI) (Ranya N. Sweis, 2024).

The diagnosis of myocardial infarction is confirmed when there is acute myocardial injury in a patient with symptoms or signs of myocardial ischemia on an electrocardiogram (ECG). Moreover, myocardial infarction is categorized into five main subtypes. Firstly, Type 1 myocardial infarction results from atherosclerotic plaque rupture which causes turbulent blood flow, platelet aggregation and coronary artery occlusion leading to myocardial ischemia and infarction. Type 2 myocardial infarction however arises from an imbalance between oxygen supply and demand without atherosclerotic plaque in which the clinical outcomes are significantly worse than Type 1, due to the age and comorbidities of the patients. Type 3 myocardial infarction usually happens in cases of sudden death before biomarker sampling can be performed. Type 4 myocardial infarction is associated with percutaneous coronary intervention (4a), stent thrombosis(4b) or in-stent restenosis (4c). Lastly, type 5 myocardial infarction often occurs after cardiac surgery. Hence, the process of the diagnosis of this disease itself is a big challenge for healthcare professionals to accurately diagnose and planning the treatment plan for their patients (Taggart et al., 2021).

In general, a comprehensive treatment plan for myocardial infarction begins with prehospital care, which includes administering oxygen, aspirin, nitrates and ensuring prompt triage to an appropriate medical facility. In the hospital, the patients will be consulted with pharmacologic therapy which consists of antiplatelet agents, antianginal drugs, anticoagulants and other medications tailored to the patient's needs. Next, reperfusion therapy is one of the vital components and may involve the use of fibrinolytics or procedures such as angiography with percutaneous coronary intervention or coronary artery bypass surgery. Next, after the hospital discharge, the treatment plan extends to rehabilitation and chronic management of coronary artery disease (Ranya N. Sweis, 2024).

In recent years, the decision tree and support vector machines (SVM) algorithms have been implemented widely for analyzing complex datasets and aiding in clinical decision-making due to its high accuracy. Compared to other algorithms implemented in similar systems, the decision tree algorithm has achieved an accuracy of 97% in brain tumor diagnosis through MRI images compared to the SVM algorithm which achieves an accuracy of 89%(Charan et al., 2022). Besides, a comparative analysis of heart disease prediction using various machine learning algorithms, the decision tree algorithms managed to achieve 85.19% accuracy compared to other algorithms such as Naïve Bayes

(74.07%), Logistic Regression (83.33%) in predicting heart disease based on patient's clinical symptoms (Rikendra et al., 2023). Besides, in order to aid increase the diagnoses accuracy, the support vector machine (SVM) can also be implemented as for its high accuracy performance in other similar systems especially for the classification of normal and abnormal electrocardiogram (ECG). For example, the support vector machine (SVM) has shown excellent performance in the classification of normal and abnormal electrocardiogram (ECG) signals by achieving 99.50% (Kumar et al., 2020).

1.2 Problem Statement

The diagnosis of diseases presents an international healthcare challenge because of the complicated nature of disease mechanisms as well as underlying symptoms despite the significant adavances of the technology (Alowais et al., 2023). The implementation of artificial intelligence (AI), especially machine learning, has the potential to revolutionize various aspects of healthcare including disease diagnosis (Alowais et al., 2023). Medical professionals worldwide face diagnosis and treatment recommendation challenges when it comes to myocardial infarction.

According to the Mantle (2021), cardiologists often face negligence claims for failing to diagnose or treat myocardial infarction properly with up to 25% of cardiac events going unrecognized. The common reasons for these claims include not suspecting myocardial infarction and misinterpretation of ECG results.

Additionally, while some patients have medical histories suggesting myocardial infarction, it can also occur without any clinical risk factors (Mantle, 2021). Thus, the challenge of accurate myocardial infarction diagnosis leads clinicians to face a decision between additional testing or hospital admission. Due to these challenges, cardiologists are less likely to be held liable for not diagnosing myocardial infarction in patients either with or without atypical symptoms.

For this reason, the availability of extensive medical data collection enables us to overcome the challenges. Hence, the decision tree and support vector machine (SVM) algorithm are chosen to be implemented to this prototype which aims to assist healthcare professionals in making accurate diagnoses and treatment plans for myocardial infarction patients.

1.3 Objective

Defining the objectives is important for guiding the research and ensuring focused progress toward achieving the goal of this project. Hence, for this project, the proposed objectives are outlined below:

i. To investigate the implementation of the decision tree and support vector machines (SVM) algorithm for medical diagnosis and treatment plan recommendation for myocardial infarction patients.
ii. To develop a prototype that will provide diagnosis and treatment plans for myocardial infarction patients based on the patient's medical history using the decision tree and support vector machines (SVM) algorithms.
iii. To evaluate the performance of the decision tree algorithm and support vector machines (SVM) in making diagnoses and recommending treatment plans for myocardial infarction patients.

1.4 Project Scope

The objective of this work involves developing a prototype using decision tree and support vector machines (SVM) to improve myocardial infarction patient diagnosis and treatment planning. The prototype uses patient data consisting of clinical symptoms alongside biomarkers and imaging results to deliver real-time evidence-based diagnosis and treatment recommendations for healthcare professionals.

i. Target user: The primary target users of this prototype are the healthcare professionals involved in diagnosis and treatment of myocardial infarction including cardiologists, emergency room physicians and clinical staff.
ii. Dataset: The two datasets that will be used in this prototype is obtained from Kaggle. The first dataset consists of 1700 data categorized in 124 demographic information columns. The second dataset consists of 982 images of normal and myocardial infarction patients' electrocardiogram (ECG) graphs.
iii. Algorithms used: The decision tree algorithms will be used for classifying patient's medical complications and Support Vector Machines (SVM) algorithms will be used to extract features from electrocardiogram graph images.
iv. Process: The process of developing this prototype consists of data collection and pre-processing from Kaggle, training and algorithms performance evaluation and developing a user-friendly interface for healthcare professionals.
v. Technology: The model will be developed using Python programming language.

1.5 Project Significance

The proposed prototype aims to improve the effectiveness and efficiency of healthcare delivery. This prototype can greatly benefit the government health departments by assisting healthcare professionals resulting in enhancing the accuracy and speed of myocardial infarction diagnosis and treatment decisions in hospitals leading to better resources utilization and improved patient care. Besides the community will also benefit from reduced related risks associated with myocardial infarction as the accurate and precise recommendations can lead to more appropriate medical interventions.

Moreover, the proposed prototype is also expected to contribute in saving costs for the healthcare sector. The accurate and efficient diagnosis and treatment of myocardial infarction will reduce the need for expensive emergency care and long-term hospitalization.

Lastly, this prototype will deliver improved life quality benefits to people who suffer from myocardial infarction. Accurate and precise treatment plans from this prototype reduce long-term health complications due to myocardial infarction and lead patients toward better health and increased activity. Furthermore, this prototype can lead to a more responsive healthcare system, not only for myocardial infarction but other medical conditions as well.

1.6 Overview of Research Framework

The research framework of this prototype consists of three phases which include the preliminary phase, design and implementation phase and testing and evaluation phase. During the preliminary phase, the preliminary study will be conducted followed by knowledge acquisition. The expected outcome for this phase is the performance of the proposed algorithms and the accuracy of their results. Next, the data collection for this prototype will also be conducted. The preliminary phase aims to fulfill the first objective of this project. Next, the design and implementation phase will be conducted which the expected outcome is the diagnosis and treatment recommender prototype for myocardial infarction using the decision tree and support vector machine (SVM) algorithm. Lastly, the testing and evaluation phase will be conducted in the expected outcome of this phase is the accurate diagnoses and treatment plan for myocardial infarction patients.

Table 1 Overview of research framework

Research Methodology	**Objective**	**Task**	**Activities**	**Deliverables**
Phase 1: Preliminary Study	**Objective 1:** To investigate the implementation of decision tree algorithm &	Literature Review	• Reading journals, articles, books and forums.	• Research background • Problem statements • Objectives • Significance

	SVM for medical diagnosis and treatment plan recommendation.		• Watching videos from YouTube on related topics and techniques.	• Potential methods/technique • Testing/Evaluation methods
		Data Pre-processing	• Data Collection • Data Cleaning	• Data acquisition
<u>Phase 2:</u> **Design and Implementation Phase**	<u>Objective 2:</u> To develop a prototype that will provide diagnosis and treatment plan for myocardial infarction using decision tree and Support Vector Machines (SVM) algorithms.	System design & algorithm development	• Define system requirements and specifications • Develop the user interface and other system components	• Detailed design document • Developed prototype for diagnosis and treatment plan recommender for myocardial infarction using Decision Tree & Support Vector Machine (SVM)
<u>Phase 3:</u> **Testing and Evaluation Phase**	<u>Objective 3:</u> To evaluate the accuracy of the decision tree and SVM in making diagnosis and treatment plan for myocardial infarction patients.	Performance evaluation	• Use a test dataset to evaluate prototype accuracy. • Calculate metrics such as accuracy, precision and recall.	• Performance evaluation report. • Performance metric analysis.

1.7 Conclusion

In conclusion, the process of accurately diagnosing and recommending treatment plans for myocardial infarction patients based on their clinical history and electrocardiogram (ECG) signals is a global challenge in the medical field. Hence, to overcome this challenge, the need for a prototype that will diagnose and recommend treatment plans for myocardial infarction patients by implementing decision tree and support vector machine (SVM) algorithm will assist the healthcare professionals in this process. The primary target users of this prototype are the healthcare professionals involved in the diagnosis and treatment of myocardial infarction including cardiologists, emergency room physicians and clinical staff. As a result, this prototype aims to improve the effectiveness and efficiency of healthcare delivery.

CHAPTER 2

LITERATURE REVIEW

This chapter provides the literature review of this project. This section delivers both an extensive definition of myocardial infarction and thorough details about recommender systems alongside discussions regarding decision trees with support vector machines (SVM).

2.1 Introduction

In recent years, machine learning algorithms have shown great performance in various domains including medical diagnostics. Among these, decision tree and Support Vector Machine (SVM) algorithms have gained significant attention for their ability to handle complex datasets. These algorithms are known for their accuracy and effectiveness making them suitable for nuanced medical diagnoses including myocardial infarction.

This literature review aims to investigate the implementation of decision tree and SVM algorithms in diagnosing myocardial infarction and recommending appropriate treatment plans. By examining the existing research, this literature review aims to increase the understanding of how these algorithms are applied in various domains, clinical settings, their performance and potential advantages. Additionally, this review will also identify the current gaps in the literature and give proper insight into the proposed recommender system itself.

2.2 Recommendation System

The recommender systems are defined as the software tools and methodologies devised to offer users recommendations that closely align with their interests and preference (Mohebbi et al., 2023). These systems essentially aim to anticipate and present the most appropriate products or services to users based on their individual preferences and constraints. These systems are extensively employed across various platforms such as online retail, streaming services and social networks to streamline the process of product selection for users.

Additionally, the recommender systems comprise several essential components including the user interface (Massoudi et al., 2021). The user interface facilitates interaction, allowing users to engage with the system and provide feedback on recommended items.

2.1.1. Types of Recommendation System

In developing a recommender system, it is essential to acknowledge there are various types available, each with its own strengths. These systems can be broadly categorized into content-based, collaborative filtering, knowledge-based and hybrid systems.

Collaborative filtering is one of the most popular types of recommender systems. The main concept of collaborative filtering revolves around identifying patterns in user behavior. For instance, if two users share similar interests and have purchased the same item, it is highly probable that they will continue to have similar preferences in the future (Laseno & Hendradjaya, 2019). This method leverages the collective preferences and behaviors of a large user base to make personalized recommendations which enhances its accuracy. By analyzing these patterns, collaborative filtering can effectively predict and suggest items that align with the user's potential preferences.

Apart from that, the recommender systems also primarily utilize into the content-based category. Content-based filtering is a technique to generate recommendations by analyzing the similarities between items, utilizing item features to infer user preferences (Turnip et al., 2017). These algorithms analyze item descriptions to identify those that might interest a particular user.

Moreover, the knowledge-based recommender system utilizes the user and product information in making recommendations. In this system, the user interest information will be used by the system to give recommendations by checking and matching it with the information that is already provided in the database. Lastly, the hybrid recommender-type system functions by combining different algorithms with the goal of achieving higher efficiency and reducing the limitation from the other techniques (Mohebbi et al., 2023).

2.1.1. Phases of Recommendation Process

The general phases of developing a recommender system begin with the data pre-processing phase. Data pre-processing is the process of cleaning and filter the data by identifying and rejecting null values, empty values and repetitive rows from the decision-making process in order to produce a high-accuracy decision (Silpa et al., 2023).

The next phase of development of a recommender system is the feature extraction from the data provided. In this phase, the unprocessed raw data will be transformed into numerical features that will be handled while retrieving the original dataset's content (Silpa et al., 2023).

Lastly, the decision-making phase involves utilizing the processed and feature-extracted data to train and deploy the recommendation model. This phase focuses on developing algorithms that can accurately predict user preferences and suggest items that the users are likely to find relevant. It includes selecting appropriate machine learning techniques, fine-tuning the model parameters, and evaluating the model's performance to ensure it meets the desired accuracy and efficiency standards. In the healthcare domain especially for myocardial infarction, the most important goal of a recommender system is to provide optimized recommendation for treatment plan by ensuring precision, reliability and accuracy based on the patient's clinical history.

2.3 Myocardial Infarction

Myocardial infarction which is also commonly known as heart attack disease is a severe medical condition resulting from the interruption of blood flow to the heart that often leads to tissue damage or cardiac death. The pathology of myocardial infarction is categorized into two main categories of ST-elevation (STE-MI) and non-ST-elevation MI (NSTE-MI). Moreover, unstable angina which is a precursor to myocardial infarction is categorized under acute coronary syndrome (ACS) (Salari et al., 2023).

2.3.1. Types of Myocardial Infarction

In general, patients receive five distinct subtypes when receiving a myocardial infarction diagnosis. Type 1 myocardial infarction develops from atherosclerotic plaque rupture creating turbulent blood flow and platelet aggregation that results in coronary artery blockage. This rupture commonly results in myocardial ischemia and infarction. Type 2 myocardial infarction arises from an imbalance between oxygen supply and demand without the involvement of an atherosclerotic plaque. Due to the age and comorbidities of the patients, type 2 myocardial infarction commonly has worse clinical outcomes than type 1. The next subtype of myocardial infarction is the type 3 myocardial infarction. This type typically occurs in cases of sudden death usually before the biomarkers can be sampled. The type 4 myocardial infarction however is associated with percutaneous coronary intervention (PCI). This type is further divided into three subtypes: 4a, related to the intervention itself; 4b happens due to stent thrombosis and 4c which results from in-stent restenosis. Lastly, type 5 myocardial infarction commonly occurs following cardiac surgery. The general process of diagnosis of myocardial infarction begins if there is an acute myocardial injury in a patient with symptoms or visible signs of myocardial ischemia detected on the electrocardiogram (ECG) (Taggart et al., 2021).

2.3.2. Myocardial Infarction Diagnosis

The ECG remains a fundamental tool in diagnosing myocardial infarction (MI) and commonly be obtained and interpreted within 10 minutes of patient presentation. Since myocardial infarction-related ECG changes can be transient, it is recommended to acquire ECGs at 15-30 minutes intervals especially if the initial reading is inconclusive (Reddy et al., 2015).

One of the primary manifestations observed on the ECG is ST elevation or ST segment as shown in **Figure 1**. This elevation is characterized by new changes observed at the J point in two contiguous leads. The magnitude of ST elevation is determined by specific criteria which in most leads, it should be at least 0.1 mV. However, in leads V2-V3, different cutoff points apply based on gender and age. For men aged 40 years or older, the cutoff is 0.2 mV while for men under 40, it is 0.25 mV and for women is 0.15 mV(Reddy et al., 2015).

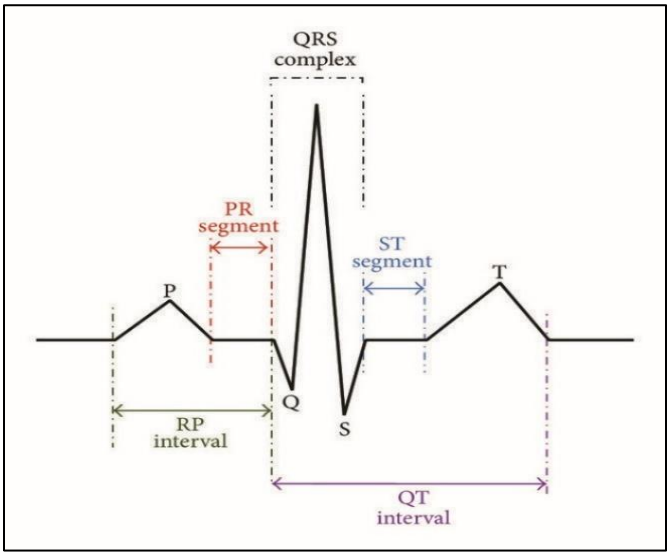

Figure 1 Normal ECG (Beyrami Enanlou & Lotfivand, 2017)

2.3.3. Myocardial Infarction Treatment

In general, for treating acute coronary syndromes (ACS) including myocardial infarction, a combination of medications will be prescribed to the patient which typically used to prevent further clotting and managing related symptoms. One of the medications includes an initial dose of 325 mg aspirin followed by a maintenance dose of 81 mg daily to minimize bleeding risks. These medications are usually given to the patients as early as possible. The next medication includes clopidogrel which is often administered with a 600 mg loading dose followed by a daily dose of 75 mg for at least a year. However, due to the higher doses of clopidogrel might increase bleeding, prasugrel and ticagrelor are some of the alternatives that can be prescribed for the patients. 60 mg loading dose followed by a 10 mg daily dose of prasugrel is proven effective in reducing ischemic events but increases major bleeding risks making it suitable for younger patients with high ischemic risk but low bleeding risk. Moreover, ticagrelor (180 mg loading, 90 mg twice daily) is also effective in reducing ischemic events and overall mortality but causes a higher risk of non-CABG-related bleeding (Reddy et al., 2015).

2.4 Decision Tree Algorithm

For this prototype, one of the proposed techniques that will be implemented will include the decision tree algorithm. Decision tree is one of the most popular supervised learning models used in multiple fields especially in machine learning, data mining and statistics. Hence, in this section, the discussion will be focussed on its mechanism and types of decision tree algorithms that can be implemented in this project

2.4.1. Operational Mechanism

A decision tree model is a structure that resembles a flowchart used for making decisions or predictions from training datasets. This model can be utilized for both categorical classification and continuous-value regression tasks. The inputs for this model are data features or attributes which can be either discrete or continuous. The structure of a decision tree consists of a root node that represents the complete dataset and initial decision as shown in **Figure 1**. The root node will be connected to internal nodes that represent the decisions or tests on attributes. These internal nodes then will consist of one or more branches that indicate the result of a decision or test which leads to another node and finally ends with the leaf nodes which also indicate the final decision or prediction (Rikendra et al., 2023).

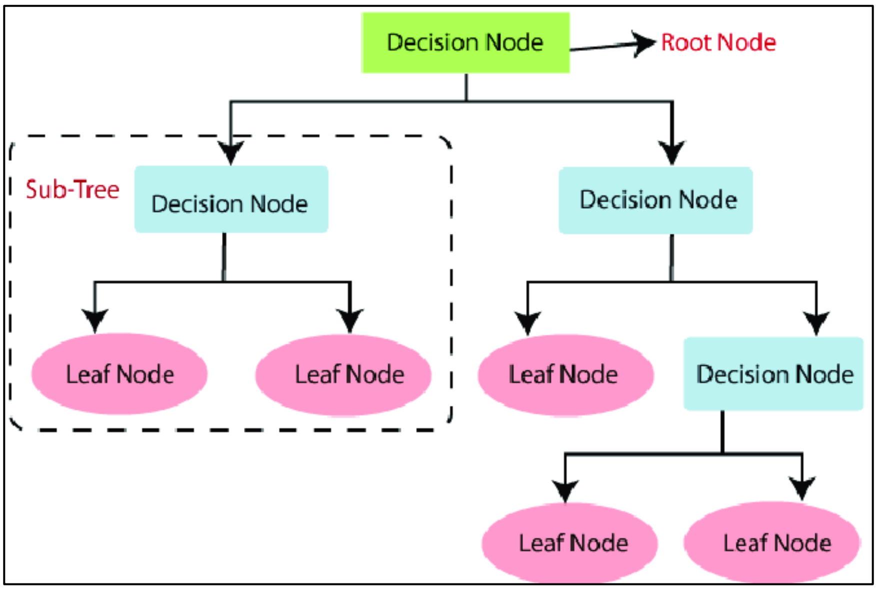

Figure 2 Structure of Decision Tree (Hafeez et al., 2021)

2.4.2. Classification and Regression Tree (CART)

Founded in the 1980s by Leo Breiman, Classification and Regression Tree (CART) is one of the most common decision tree techniques implemented in healthcare for diagnosing and suggesting treatment plans. In this technique, a decision tree will be constructed using historical data with pre-defined labels and a regression tree for the classification model(Aryuni et al., 2021). In the healthcare field, it is essential to consider various parameters and combinations within the CART algorithm which include the pruning method, the ordering of tree construction and the dataset sampling approach (Gozali, 2023). These parameters and combinations are fundamental for the CART to achieve high accuracy in prediction or classification.

For instance, the CART model was implemented in a model to predict sudden cardiac death based on a patient’s electrocardiogram (ECG) and clinical data.

In this project, the CART model was trained with 50% of the training dataset and 50% of the testing dataset. In making the prediction, each node or leaf will be tested until a condition is met, hence a new one will then be tested until the model reaches the predetermined conditions or meets the thresholds (Pereira & Karia, 2018).

Moreover, in a prediction model developed for cervical cancer, Classification and Regression Trees (CART) were built in order to estimate the most likely value of an independent variable given a dataset that contains continuous dependent variables and categorical or predictive elements (Singh et al., 2023). Hence, there would be only two possible outcomes for each decision node in the decision tree produced by CART. These outcomes were constructed by dividing the record into groups in a recursive process based on how similarly the target qualities would be evaluated. Moreover, this algorithm will grow the tree that predicts if the cervix is malignant or else (Singh et al., 2023).

2.4.3. Random Forest

Another commonly used decision tree model in the healthcare field includes the random forest model. The random forest model is one of the most popular supervised learning methods that employs both classification and regression techniques with the decision trees constructing a random forest. By utilizing Bootstrap aggregating, the random forest model managed to reduce variance effectively, thereby mitigating the risk of overfitting. Bootstrap aggregating or bagging involves the process of randomly selecting features with replacements. The root node selection in the random forest is chosen based on the highest votes. Compared to other models, random forest refrained from pruning or reducing branches which resulted in high prediction accuracy (Lutimath et al., 2021).

In predicting myocardial infarction or heart-related diseases, the random forest model is widely used due to its high accuracy. For instance, the random forest model was implemented in a model developed to detect myocardial infarction with various test scenarios. Initially, the random forest model design involved two crucial stages which are the data processing and classification. The data processing stage consists of cleaning the patient's medical record and scaling the data values. Following this, the classification stage utilizes the processed data to identify patients who show potential indicators of myocardial infarction (Rambe & Mandala, 2023). After these two stages have been completed, the random forest model will be implemented in two different scenarios: the default random forest model and hyperparameter tuning on the random forest model. In the initial scenario, the random forest model was utilized with its default settings to detect myocardial infarction with no modifications made to the algorithm. However, in the second scenario, the random forest model was modified with customized hyperparameters tailored to the specific data in order to enhance the model's performance. As a result, in detecting myocardial infarction, the default model achieved an accuracy of 82.61% while the hyperparameters-tuned model achieved an accuracy of 89.13%, representing an increase of 6.52% (Rambe & Mandala, 2023).

2.5 Support Vector Machine Algorithm

Support Vector Machine (SVM) is a powerful supervised machine learning model that is frequently utilized for binary classification which is ideal for the medical diagnosis process. Compared to other available algorithms, SVM stands out due to its strong mathematical background. The main idea in the implementation of SVM is to search for a hyperplane that best separates the data into two classes. In the field of medical diagnosis, SVM is one of the algorithms that has been widely used due to its capability to handle high-dimensional data and its high accuracy.

In the context of myocardial infarction, the SVM algorithm has been widely implemented to differentiate normal and abnormal ECG. The SVM works by employing kernel functions to transform a low-dimensional space into a higher-dimensional space which is effective in converting non-separable problems into separable ones. In the classification of data like ECG signals, the commonly used kernel functions include Gaussian, polynomial and radial basis (RBF) (Kumar et al., 2020).

2.5.1. SVM Linear

While SVMs are widely known for their ability to handle non-linear classification through various kernel functions, one of the most effective kernels implemented in many practical applications includes the linear kernel, also known as SVM linear. Compared to other kernels, the linear SVM is much simpler and faster to compute which makes it more effective for large datasets. For example, SVM linear was one of the algorithms implemented in extracting features and classifying cardiac ventricular arrhythmias based on ECG signals. This classification would eventually be the reference to detect heart-related diseases for the subject. In this model, the normalized dataset was fed into various classifier algorithms which includes Naïve Bayes, SVM linear and decision tree (Rajendran et al., 2021).

2.6 Implementation of Decision Tree and Support Vector Machines (SVM) in Various Domains

Decision tree algorithms are widely recognized for their effectiveness in various recommender system applications due to their simplicity, interpretability and capability to handle both categorical and numerical data resulting in precise decisions. This flexibility makes the technique one of the most suitable choices for recommender systems across various domains.

Table 2 Implementation of algorithms in various domains

No.	Research Paper	Dataset		Algorithm	Result
		Source	**Type**		
1.	MTRecS-DLT: Multi-Modal Transport Recommender System using	Context-Aware Multi-Modal Transportation Recommendation challenge	Text	Convolutional Neural Networks (CNN) models,	F1-Score: 0.68898702.

	Deep Learning and Tree Models(Abedalla et al., 2019)			Gradient Boosting Decision Trees (GBDT)	
2.	Agriculture Recommender System for Precision Farming using Machine Learning(ARS) (Kolikipogu et al., 2023)	Kaggle	Text	Random Forest, Decision Tree, SVM	Accuracy (SVM): 94.5% Accuracy (Decision Tree): 92.3% Accuracy (Random Forest): 95.9%
3.	ENSEMBLED CROPIFY – Crop & Fertilizer Recommender System with Leaf Disease Prediction (Priya et al., 2023)	Kaggle	Text	Decision Tree, Random Forest, Support Vector Machine (SVM), Logistic Regression and Gaussian Naïve Bayes	Accuracy: 99.67%
4.	Food recommender system based on weighted ingredients, body mass index and allergies; using the Random Forest algorithm (Martínez-Gorospe et al., 2021)	Not mentioned	Text	Random Forest	Precision: 97.7381%
5.	Career Recommender System Using Decision Trees	Kaggle	Text	Decision Tree	Accuracy: 85.81%

	(Massoudi et al., 2021)				
6.	College Student Employment Management Recommendation System Based on Decision Tree Algorithm (Wang, 2022)	Not mentioned	Not mentioned	Decision Tree	Successful prediction for 45 students.
7.	Recommender system for ubiquitous learning based on decision tree (Guabassi et al., 2016)	Not mentioned	Not mentioned	Decision Tree	This recommender system was presented to the target user for the collection of the learner's feedback as this helps reduce the likelihood of suggesting learning objects that do not interest the learner
8.	Feature Selection based on Random Forests and its Application to a Modeling of Sewage Treatment Plant (Kimura et al., 2019)	Daily reports of the sewage treatment plan in Tamano city	Text	Random Forest	Higher sensitivity in feature selection but a lower specificity.
9.	Wart Treatment Selection with a Decision Tree-Based Approach (Yanik & Cömert, 2019)	Ghaem Hospital in Mashhad	Text	Decision Tree	ROC Curve (Cryotherapy): 0.9821 ROC Curve (Immunotherapy): 0.9507
10.	A Decision Tree Based Method for	WiseHealthcare TECHNOLOGY	Text	Decision Tree	Precision: 65%

	Treatment Therapy of HCC (Zhao et al., 2019)	(SHANGHAI) Co.Ltd			
11.	Decision Tree Algorithms for Predictive Modeling in Breast Cancer Treatment (Kutrani & Eltalhi, 2022)	Oncology department at BMC (Benghazi Medical Center)	Text	J48, CART, Random Forest	ROC Curve: 0.963

2.6.1. Non-healthcare Domains

For instance, in transportation services, the decision tree algorithm was implemented in developing a multi-modal recommender system. Multimodal transportation consists of a complex network system that offers individuals a variety of route options involving multiple combinations of transport modes such as walking, cycling, driving, trains and buses. Hence, the multi-modal recommender system aims to provide the most optimized transportation mode by considering user preferences like cost, distance and traffic congestion. By utilizing the public dataset provided by the Context-Aware Multi-Modal Transportation Recommendation Challenge 2019, this model was constructed using the combinations of Convolutional Neural Networks (CNN) models, Gradient Boosting Decision Trees (GBDT) model and other deep learning models (Abedalla et al., 2019). The Gradient Boosting Decision Trees (GBDT) model is an ensemble algorithm that iteratively trains new models to correct the errors of previous models. As a result, the combinations of these models between Convolutional Neural Network (CNN) and eXtreme Gradient Boosting (XGBoost) successfully outperformed the deep learning models by achieving the weighted F1-score of 0.68898702. Hence, it shows that the decision tree model is one of the most effective models for recommending the most suitable transport mode due to its robust performance.

Moreover, the decision tree algorithm can also be implemented in a food recommender system based on weighted ingredients, body mass index and allergies. In this recommender system, the decision tree model, specifically the random forest model has been trained using the data on health and users' level of ingredient consumption which resulting the most appropriate list of ingredients. Compared to other models implemented in this system, the Random Forest algorithm excels by achieving a precision of 97.7381% in recommending suitable ingredients based on the input data of the users (Martínez-Gorospe et al., 2021).

Besides, in the agriculture domain, the decision tree algorithm is also one of the power tools utilized in the crop and fertilizer recommender system with leaf disease prediction. In this system, the Random Forest algorithm was implemented as this model aims to produce higher accuracy while the Decision Tree algorithm on the other hand was mainly used in the classification (Priya et al., 2023). There were also a few models implemented in this system such as Support Vector Machine (SVM), Logistic Regression and Gaussian Naïve Bayes. The first phase of this system is to recommend a suitable fertilizer for a particular crop based on the crop's needs for nitrogen, potassium and phosphorus. Hence, the SVM algorithm and Random Forest algorithms were used for classification and these algorithms managed to provide the features of the fertilizer recommendation. Next, to differentiate between healthy and non-healthy leaf, the plant leaf disease prediction was constructed by utilizing the CNN algorithm on the dataset of leaves images from Kaggle datasets. Finally, the experimental results of this system show that the Random Forest algorithm managed to achieve the highest accuracy of 99.67% followed by the SVM algorithm with 99.45% accuracy indicating their effectiveness in this crop and fertilizer recommender system (Priya et al., 2023).

Furthermore, in a similar field, leveraging Machine Learning including SVM, Decision Tree and Random Forest also helps in developing an agriculture recommender system for precise farming. Precision farming is a farming technique that utilizes cutting-edge tools in optimizing crop yield by reducing environmental impact and efficient use of resources. Hence, the adoption of these advanced techniques including machine learning is not only effective but also useful for making accurate decisions for precision farming based on the soil composition and climate characteristics (Kolikipogu et al., 2023). These models are trained with the crop recommender dataset which consists of unique features including nitrogen content, phosphorus content, potassium content, temperature, relative humidity, soil pH value and rainfall rates. As the result of training and testing the models, the accuracy percentage for the SVM model is 94.5% while the Decision Tree Classifier achieves 92.3% and the Random Forest Classifier achieved 95.9% accuracy (Kolikipogu et al., 2023).

Nowadays, numerous career options available in computer science specifically have become a challenge to individuals in identifying their skills and defining their professional objectives, leading to uncertainty about their desired career path (Massoudi et al., 2021). Hence, it is crucial for individuals to assess their professional skills, interests and certifications to make well-informed career decisions. Thus, a career recommender system was developed by integrating the decision tree algorithm to analyze an individual's strengths and provide guidance in selecting an appropriate career path. By entering candidate information into the system, the user can efficiently determine the most suitable roles for each individual, providing an optimized career decision for the particular person. In this system, the decision tree model is trained after dividing the data into two groups: a training set and a test set. The Decision Tree Classifier from the sklearn package then is utilized for both training and testing the model. After the training process finished, the data was processed and evaluated using both the Gini and Entropy methods. The model's performance, including the confusion matrix and accuracy is assessed for various depths ranging from 1 to 10. For the result, the accuracy of decision trees based on the information analysis method was 85.81% indicating the efficiency of the system in recommending suitable careers for a particular individuals (Massoudi et al., 2021).

Moreover, the decision tree algorithm specifically the random forests model is also effective in feature selection for modeling sewage treatment plants. A sewage treatment plant facility designed to eliminate contaminants from wastewater through a combination of physical, chemical and biological processes producing treated wastewater. This process also generates sewage sludge which is composed of water with a small fraction of solid material (Kimura et al., 2019). Hence, the integration of the random forest models aims to search for an optimized technique for controlling the sewage treatment processes resulting in reduced operating costs for the sewage treatment plan. In achieving this purpose, the algorithm was implemented to predict the water content within the sludge cake based on the information retrieved and providing the best-recommended model.

Additionally, the rapid advancement of mobile, wireless communication and sensor technologies has opened up new opportunities. One of the opportunities includes ubiquitous learning which allows for learning anywhere and anytime. However, to ensure the learners have an optimal learning experience, it is vital to consider factors such as learner characteristics and context. Hence, it requires the need of a recommender system for ubiquitous learning that leverages learner context information and employs a decision tree model for the decision-making process (Guabassi et al., 2016). The development of this recommender system consists of two main components. The first component is Context-based Data Mining which explores all potential combinations of various learning object options such as text, audio or video and contextual information that includes mobility, noise, luminosity and connectivity. The second component involves the generation of the recommendation algorithm using a decision tree. This process converts samples into a tree structure from which adaptation rules are derived. Finally, this recommender system was presented to the target user for the collection of the learner's feedback as this helps reduce the likelihood of suggesting learning objects that do not interest the learner (Guabassi et al., 2016).

Moreover, with the continuous expansion of colleges and universities has increased the number of graduates which leads to significant employment pressure for both students and society. Hence, to solve this problem, a recommendation system for managing college student employment using a decision tree algorithm (Wang, 2022). The process of developing this system includes constructing a decision tree and using it to derive classification rules resulting in an employment prediction model. By using the data obtained from new students in the School of Computer Science and Engineering, the model was trained and tested to predict employment outcomes. According to the experimental result, the decision tree model successfully predicts the employment level of fresh graduates including 45 graduates who found employment in large-scale enterprises. Hence, it indicates the integration of decision tree algorithm in this recommendation system has been effectively applied to support the employment prospects of students (Wang, 2022).

2.6.2. Healthcare Domains

Furthermore, the decision tree and SVM algorithms are also being widely implemented in the healthcare domain to build recommender systems that will provide optimized treatment plans for particular patients. The implementation of these algorithms has assisted healthcare professionals globally in making accurate and precise diagnoses and eventually building treatment plans that help make a better life for their patients.

For example, in developing a recommender system for wart treatment selection, the decision tree algorithm excels in providing the best treatment plan. Warts are a skin disease caused by human papillomaviruses (HPVs) that can grow in all parts of the human body. The most commonly known warts are plantar warts which mainly grow on the soles of the hands and feet. Cryotherapy and immunotherapy are two of the most commonly applied methods in wart treatment. However, the most effective treatment is commonly based on the patient's clinical signs which require the need of a recommender system using a decision tree-based approach (Yanik & Cömert, 2019). During the development of this system, there were two datasets used to train the model which were collected from the dermatology clinic of Ghaem Hospital in Mashhad. The attributes collected for the decision consist of gender, age, number of warts, type of warts, time elapsed before treatment and the surface area of warts which would be providing the optimized treatment result. Next, with the selected treatment techniques, the two different models with decision trees were trained and the best parameters were observed for the best ROC area while the best decision was created for the decision of these two therapies. As a result, the area under the ROC Curve for Cryotherapy is 0.9821 while the area under ROC Curve for Immunotherapy is 0.9507 indicating that the decision tree managed to provide accurate treatment selection for each model (Yanik & Cömert, 2019).

Moreover, the proposed algorithms are also widely integrated into systems that used to provide optimized treatment plans including liver cancer. In China, liver cancer ranks as the fourth most common cancer and has become the third leading cause of cancer-related deaths. 85%-90% of liver cases were caused by Hepatocellular carcinoma (HCC). The early-stage HCC can be diagnosed through subtle symptoms that include intermittent or continuous pain in the hepatic region, accompanied by fever, anorexia, indigestion, nausea, diarrhea and weight loss. However, due to the disease's rapid progression to the advanced stage, the symptoms in this stage are usually characterized by intrahepatic or distant metastases, jaundice, bleeding tendencies, hepatic encephalopathy and the deterioration of liver and kidney functions. Hence, accurate HCC staging is crucial for treatment therapies using a decision tree-based method (Zhao et al., 2019). Consequently, the decision tree integrated into this system has achieved more than 65% average consistent rate and able to operate under data-missing situations (Zhao et al., 2019).

Additionally, in constructing a predictive model for breast cancer treatment, the decision tree algorithms including J48, CART and Random Forest were implemented. The predictive model was developed using these machine learning classifiers in order to provide accurate correct decision making especially for treatment plan for breast cancer patient. Using the dataset obtained from the Oncology department at Benghazi Medical Center (BMC), the model was trained and tested using WEKA software. As a result, using the ROC curve as the indicator of the ability of the classifiers, the Random Forest model achieved the highest ROC area for both the training and testing datasets which are 0.94 and 0.963 (Kutrani & Eltalhi, 2022).

2.7 Similar Works

One of the key steps in literature review involves the process of reviewing similar works. Reviewing similar works is important as it provides a foundation for understanding current advancements and gaps in this domain. By examining previous research on myocardial infarction diagnosis and treatment recommendation systems using algorithms like decision tree and SVM, the relevant information such as dataset, algorithm's performance and gaps in the existing research can be identified.

Table 3 Similar Works

No.	Research Paper	Dataset		Algorithm	Result	ECG/ Medical History	Treatment Recommendation
		Source	Type				
1.	Prediction Model of Heart Attack Based on Optimized Decision Tree Algorithm (Liu et al., 2023)	Database of the University of California, Irvine, USA	Text	Logistic Regression, Decision Tree	Accuracy: 80.23%	Medical history	Not available
2.	Heart Attack Prediction Using Artificial Neural Networks (Bharathi et al., 2023)	UCI ML Dataset	Text	Artificial Neural Network (ANN)	Accuracy: 98.33%	Medical history	Not available
3.	ECG Based Early Heart Attack Using Neural Networks (Kumar et al., 2022)	UCI ML Dataset	Text	Convolutional Neural Network (CNN), Artificial Neural Network (ANN)	Accuracy: 98%	ECG	Not available
4.	Machine Learning based selection of Myocardial Complications to Predict Heart Attack (Saxena et al., 2022)	UCI ML Dataset	Text	Support Vector Machine (SVM), Logistic Regression	Accuracy: 95.15%	Medical history	Not available

5.	Prediction and Analysis of Heart Attack using Various Machine Learning Algorithms (Sharma, 2023)	Kaggle	Text	Support Vector Machine (SVM), Random Forest, KNN, X Gradient Boosting	Accuracy (SVM): 91.85%	Medical history	Not available
6.	Heart Attack Probability Analysis Using Machine Learning (Shanbhag et al., 2021)	Kaggle	Text	KNN, Decision Tree, Random Forest, Logistic Regression, Support Vector Machine	Accuracy (SVM): 85.7%	Medical history	Not available
7.	Heart Attack Detection using Machine Learning (Prakash et al(Taggart(Prakash et al	Kaggle	Text	Logistic Regression, Decision Tree, Support Vector Machine (SVM)	Accuracy: 0.87	Medical history	Not available
8.	Prediction of Heart Disease using Random Forest (Lutimath et al., 2021)	UCI repository	Text	Decision Tree, Random Forest	Mean Absolute Error (MAE): 0.63 Mean Squared Error (MSE): 0.72	Medical history	Not available

					Root Mean Squared Error (RMSE): 0.85		
9.	Design and Analysis of Heart Attack Prediction Using ML (Manoj et al., 2023)	Not mentioned	Not mentioned	Logistic Regression, MLP Classifier, Random Forest Algorithm, CatBoost Regression	Accuracy: 98.84%.	Medical history	Not available
10.	Classification of Normal and Abnormal ECG signals using Support Vector Machine and Fourier Decomposition Method (Kumar et al., 2020)	MIT-BIH arrhythmia dataset	Text	Support Vector Machine (SVM), Fourier Decomposition Method	Accuracy: 99.50%	ECG	Not available
11.	Classification of heart disease from ECG signals using Machine Learning (Rajendran et al., 2021)	MIT-BIH Arrhythmia Database	Text	Naïve Bayes, Fine Tree, Medium Tree, Coarse Tree, Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian	Accuracy (Linear SVM): 99.4%	ECG	Not available

				SVM, Boosted Trees, Bagged Trees			
12.	Deep Learning approach for MI detection using SVM-based classifier (Bera, 2024)	Physionet	Images	Support Vector Machine	Accuracy: 97.37%	ECG	Not available
13.	Study on survival prediction of patients with heart failure based on support vector algorithm (Sang et al., 2020)	Medical record data set from Faisalbad Cardiology Institute	Text	Support Vector Machine, Random Forest, Decision Tree, KNN, XGBoost	Accuracy (SVM): 85.71%	Medical history	Not available
14.	Drug Recommendation System in Medical Emergencies using Machine Learning (Silpa et al., 2023)	Not mentioned	Text	Decision Tree, Logistic Regression, Random Forest, Naïve Bayes	Accuracy: 99.3%	Not available	Available
15.	Myocardial Infarction Detection as an Element of Preventive Medicine with Random Forest (Rambe & Mandala, 2023)	Kaggle	Text	Random Forest	Accuracy: 89.13%	Medical history	Not available
16.	Coronary Artery Disease Prediction Model using	Kaggle	Text	Classification And Regression Tree	Accuracy: 88.3%	Medical history	Not available

	CART and SVM: A Comparative Study (Aryuni et al., 2021)			(CART), Support Vector Machine (SVM)				
17.	Heart disease Prediction using Machine Learning (Ibrahim et al., 2023)	Kaggle	Text	K-Nearest Neighbours (KNN), Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Gradient Boosting	Accuracy :0.86.	Medical history	Not available	
18.	Machine Learning and Deep Neural Network Techniques for Heart Disease Prediction (Rahman et al., 2022)	UCI repository	Text	K-Nearest Neighbour (KNN), Naïve Bayes, Support Vector Machine, AdaBoost, XGbBoost, Random Forest, Decision Tree, Logistic Regression	Accuracy: 0.811	Medical history	Not available	
19.	Diagnosis of Left Ventricular Hypertrophy from ECG Signals Based on CCS Methodology using SVM (Mahin & Ahmad, 2021)	Physionet	Images	Support Vector Machine (SVM)	Accuracy: 80%	ECG	Not available	

20.	Machine-Learning-Based Prediction Models of Coronary Heart Disease Using Naïve Bayes and Random Forest Algorithms (Bemando et al., 2021)	UCI repository	Text	Naïve Bayes, Random Forest	Accuracy: 74.8%	Medical history	Not available

Based on **Table 3**, there are 19 research papers with related work on the diagnosis of myocardial infarction using various algorithms and techniques. Out of the 19 papers, it was stated in 6 research papers that the dataset for myocardial infarction complications was obtained from a public source which is the UCI repository dataset. However, the most used dataset for myocardial infarction diagnosis is obtained from Kaggle indicating that this public source dataset is more reliable and easier to obtain. These datasets were mostly provided in the text format and commonly used for training of classifier methods. Moreover, two research papers used the MIT-BIH arrhythmia dataset which consists of an electrocardiogram (ECG) in text format. However, it is stated in two research papers that the diagnosis of heart attack can also utilize the dataset of electrocardiogram (ECG) images. Hence it shows that most of the research papers managed to develop models that made diagnosis of myocardial infarction either based on textual data or image data specifically ECG.

In diagnosing myocardial infarction based on the complications or through ECG datasets, there were various methods implemented in various models including logistic regression, decision tree, KNN, artificial neural network (ANN), convolutional neural network (CNN), Naïve Bayes, random forest and also SVM. Out of 19 research papers regarding the diagnosis of myocardial infarction, it clearly shows that the SVM algorithm excels in 8 research papers by achieving the highest accuracy in feature extraction of the datasets provided. Moreover, the decision tree and random forest model also showed a very excellent performance in detecting myocardial infarction throughout various textual formatted data. Plus, in the context of the recommender system, Silpa et al. also mentioned in their research paper that the decision tree also achieved the highest accuracy in the drug recommendation system in medical emergencies using machine learning showing the capability of this algorithm to be implemented in a recommender system for the healthcare domain.

However, regarding the treatment recommender system for myocardial infarction, it is evident that there is a significant gap in the research literature on this topic. This absence of comprehensive studies underscores the need for innovative research and development in this area to improve patient outcomes and personalize treatment plans.

2.8 The Implication of Literature Review

The comprehensive literature analysis has provided valuable insights with significant implications for both theoretical frameworks and practical applications within the proposed techniques which are the decision tree algorithm and support vector machine (SVM) algorithm. The literature review explains on the mechanism of the decision tree algorithm and support vector machine (SVM) algorithm and the implementation of these algorithms throughout various domains. Due to the simplicity of the mechanism of the decision tree algorithm, this technique is widely used in various domains to provide accurate recommendations based on the dataset given. For the SVM algorithm, it is also significant especially for feature extraction from complex data.

In the domain of myocardial infarction, these algorithms are beneficial as the combination of these algorithms is expected to be able making accurate and precise diagnoses based on the patient's clinical history and ECG provided. Hence, holds essential value for patients because it supports immediate accurate diagnosis followed by correct treatment preparation while saving lives but also critical for reducing long-term health complications and improving overall quality of life.

Overall, the implementation of these algorithms within the proposed prototype is significant as they perform precise diagnostic assessments through complete patient data reviews leading to prompt and successful treatment interventions. Furthermore, these algorithms demonstrate excellent capabilities because they can process complex data while continuing to maintain their effectiveness in the future.

2.9 Conclusion

In conclusion, this literature review presents an extensive analysis on the significant implementation steps to utilise Decision Tree and Support Vector Machine (SVM) algorithms to diagnose and recommend treatments for myocardial infarction patients. . The discussion on the Decision Tree algorithm covered its operational mechanism in complex decision-making processes especially for myocardial infarction based on the patient's clinical history. Furthermore, the Classification and Regression Tree (CART) and Random Forest algorithms provided insights into more sophisticated approaches with the Decision Tree framework. Similarly, the SVM algorithm which focus on SVM Linear illustrated its effectiveness in handling high-dimensional and non-linear data, which in the context of myocardial infarction regarding the ECG signals. Hence, throughout the implementation of these algorithms in various domains revealed their effectiveness either in non-healthcare or healthcare to solve complex problems, which reinforces their potential in the proposed recommender system for myocardial infarction. Plus, the analysis of similar works in the proposed domain also provided valuable perspectives on the success and limitations of existing systems. As a result, the insights gained from this literature has provided valuable information that will act as guidance for the design and implementation of this prototype.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter explains the methodology that is being implied in this project. Other than that, this chapter also will show the current phase that being used in the project development. Moreover, by completing this chapter, the specification of the system requirement could be determined to start and complete the project.

3.1 Overview of Research Methodology

3.1.1 Details of Research Framework

For this prototype, the research framework is structured into three distinct phases as shown in **Table 4**: the preliminary phase, the design and implementation phase and the testing and evaluation phase. Each phase was constructed to fulfil the project's objectives, ensuring a comprehensive approach to developing the prototype using the proposed algorithms.

The initial phase, also known as the preliminary study phase is very important as it provides the groundwork for the entire research project. The primary objective of this phase is to investigate the implementation of the Decision Tree and Support Vector Machine (SVM) for medical diagnosis and treatment plan recommendations. This phase involves a comprehensive study to gather information on how these algorithms can be effectively applied in the medical field and their performance in similar works.

Hence, to achieve this objective, a literature review activity is conducted. This review includes reading a wide range of journals, articles, books and forums that discuss the implementation of these algorithms in related works and their performance throughout the systems. This phase also includes watching educational videos from YouTube on related topics and techniques to provide visual demonstrations and practical knowledge which help to enhance the understanding of complex concepts and real-world applications.

Another critical activity in this phase also involves the data pre-processing. This involves the collection and cleaning of data that will be used to develop and test the algorithms.

As a result, the expected deliverables from this phase include a detailed research background that provides context and justification for the study, clear problem statements, specific objectives and the significance of this project. Moreover, during this phase, the possible techniques equipped with testing/evaluation methods were also decided. Other than that, the clean collected data were also acquired.

Next, the second phase also known as the design and implementation phase is crucial to achieving the second objective of this project which is to develop a prototype that will provide an accurate diagnosis and treatment plan for myocardial infarction. Hence, in this phase, the system design and algorithm development will be conducted. This task includes the process to define system requirements and specifications and developing the user interface and other related prototype components. As a result, the expected deliverables include a detailed design document and the development of a prototype for diagnosis and treatment plan recommender for myocardial infarction using Decision Tree & Support Vector Machine (SVM) algorithms.

Lastly, the final phase of this project is the testing and evaluation phase. The objective of this phase to evaluate the accuracy of the Decision Tree and Support Vector Machines (SVM) algorithms in making diagnosis and treatment plans for myocardial infarction patients. This phase involves the performance evaluation using accuracy, precision and F1-recall metric.

Table 4 Detailed research framework

Research Methodology	Objective	Task	Activities	Deliverables
Phase 1: Preliminary Study	**Objective 1:** To investigate the implementation of decision tree algorithm & SVM for medical diagnosis and treatment plan recommendation.	Literature Review	• Reading journals, articles, books and forums. • Watching videos from YouTube on related topics and techniques.	• Research background • Problem statements • Objectives • Significance • Potential methods/technique • Testing/Evaluation methods
		Data Pre-processing	• Data Collection • Data Cleaning	• Data acquisition
Phase 2: Design and Development Phase	Objective 2: To develop a prototype that will provide diagnosis and treatment plan for myocardial infarction using decision tree and Support Vector Machines (SVM) algorithms.	System design & algorithm development	• Data training for decision tree using medical history dataset. • Data training for support vector machines (SVM) using ECG graph image dataset.	• Trained decision tree model. • Trained Support Vector Machines (SVM) model.
			• Define system requirements and specifications • Develop the user interface and other system components	• Detailed design document • Developed prototype for diagnosis and treatment plan recommender for myocardial infarction using Decision Tree &

				Support Vector Machine (SVM)
<u>Phase 3:</u> Testing and Evaluation Phase	<u>Objective 3:</u> To evaluate the accuracy of the decision tree and SVM in making diagnosis and treatment plan for myocardial infarction patients.	Performance evaluation	• Use a test dataset to evaluate prototype accuracy. • Calculate metrics such as accuracy, precision and recall.	• Performance evaluation report. • Model performance metrics and analysis.

3.2 Preliminary Phase

3.2.1 Literature Study

The literature study phase involves the process of knowledge acquisition related to the proposed project regarding various parts. This phase is important for understanding the current state of research and existing similar works within the relevant area of interest and domains. By reviewing existing research papers, the gaps in current knowledge can be identified hence establishing the theoretical framework and comprehensive understanding. Hence, through this process, a solid foundation for the proposed project can be established through knowledge acquisition from existing research, relevant theories and methodological theories.

Conducting a literature study involves several key steps to ensure a comprehensive and systematic knowledge acquisition from existing research. The first step is to define the problem statement and objectives of the proposed project, determining the area of interest and domain. Next, using an academic database such as IEEE Explore, Google Scholar etc., the process of searching for relevant literature through journals, articles, books and forums. This step includes identifying keywords and search terms that align with the research questions such as further explanation regarding the selected domain and possible algorithms to be implemented in this project. Once the literature is gathered, the next step taken is to critically evaluate the findings, focussing on methodologies, algorithm's performance results and theoretical contribution.

3.2.2 Data Pre-processing

Data pre-processing which consists of data collection and cleaning process is a crucial step in developing a prototype. Proper data pre-processing ensures the accuracy of the prototypes by eliminating errors and inconsistencies.

3.2.3 Data Collection

During the data collection process, the data were collected from reliable public sources such as the Kaggle website. For this proposed project, the datasets were chosen based on the similarities to existing works obtained during the literature review to ensure alignment with established methodologies and research objectives. Hence, there were two distinct types of datasets that will be used in this project. The first dataset comprises the ECG images to facilitate the application of Support Vector Machines (SVM) in extracting features from the image and classifying the images. Next, the categorical dataset was also chosen to facilitate the Decision Trees algorithm to enable the extraction of decision rules in making diagnoses based on the user's clinical data. These datasets will be integrated by combining the ECG classification results and the patient clinical data to allow for more precise risk assessment process.

For the first dataset, it contains 928 ECG images that are categorized into four classes: Normal, Myocardial Infarction, Patient with History of Myocardial Infarction and Abnormal Heartbeat as shown in **Figure 3**, **Figure 4**, **Figure 5** and **Figure 6**.

Moreover, the second dataset consists of 1700 patients' data. This dataset contains 110 input attributes that which consists of binary and numeric attributes as shown in **Table 10**.

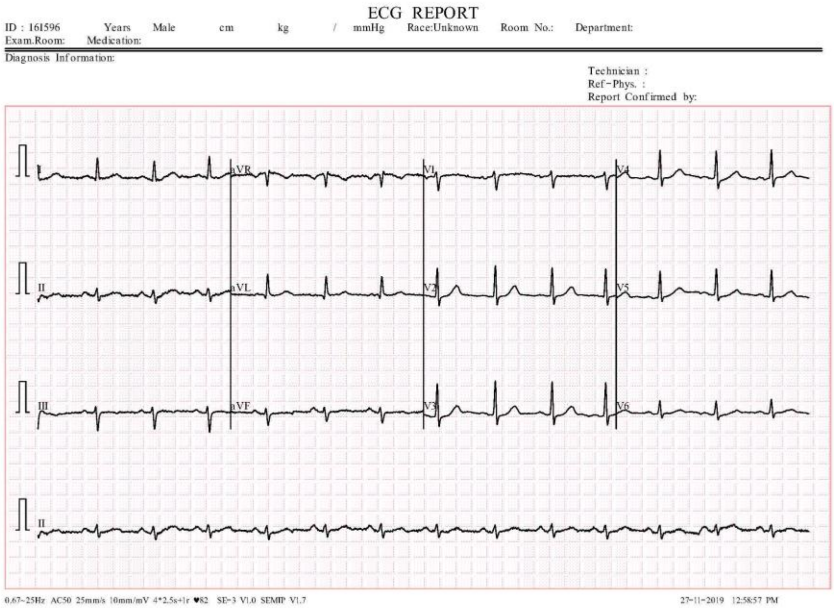

Figure 3 Example of Normal ECG

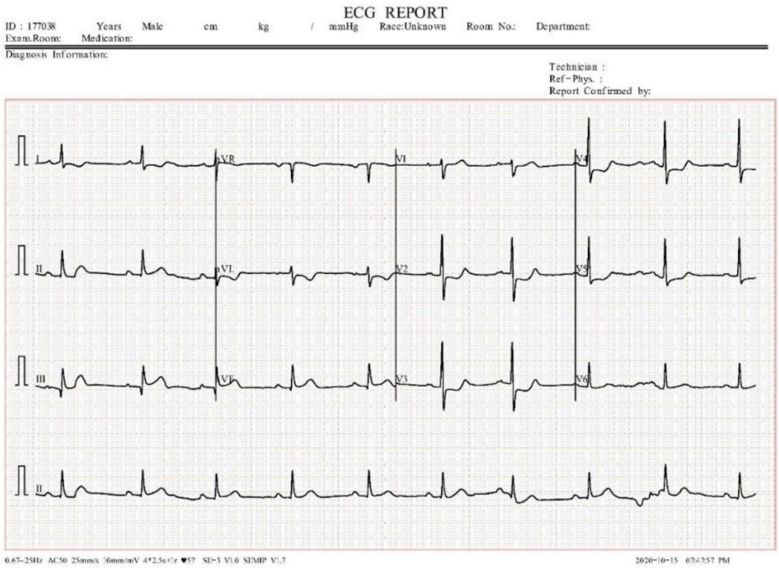

Figure 4 Example of Myocardial Infarction ECG

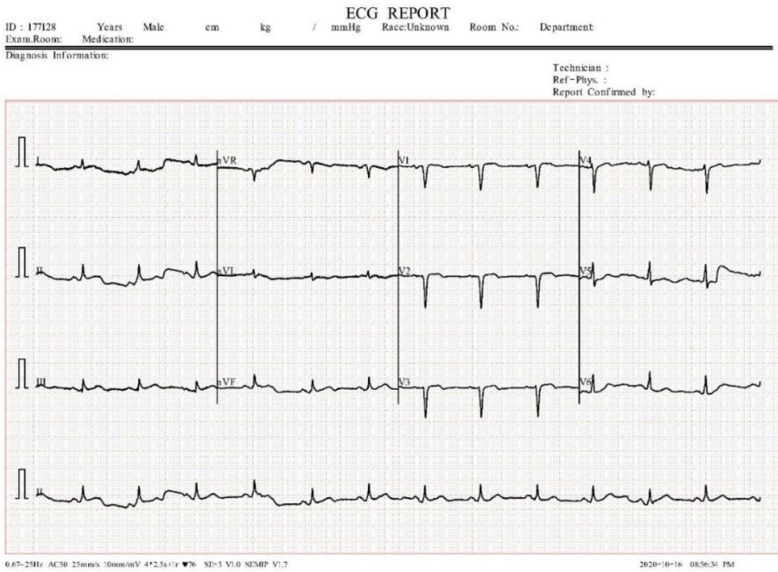

Figure 5 Example of a Patient with History of Myocardial Infarction ECG

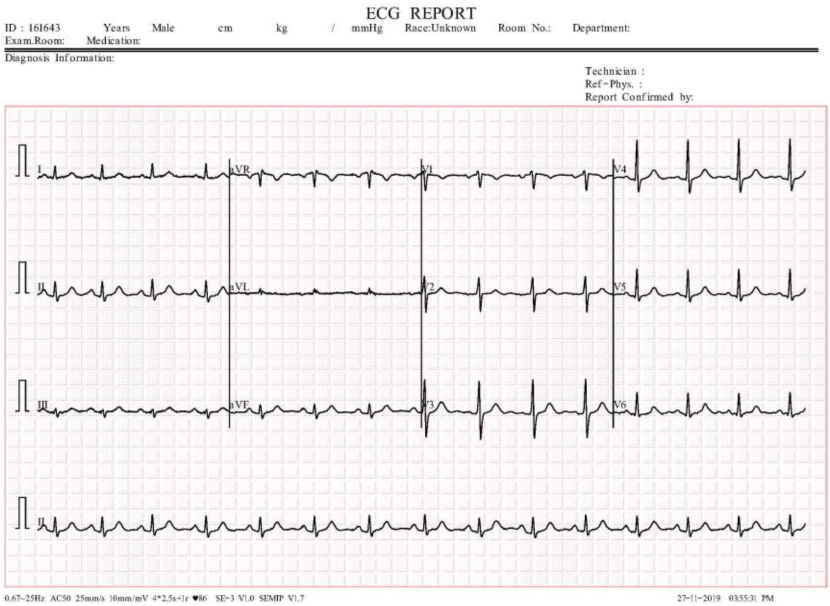

Figure 6 Example of Abnormal Heartbeat ECG

3.2.4 Data Cleaning

After the desired datasets are chosen, the next key step in pre-processing is to clean the dataset. There are several key steps involved in data cleaning. One of the key steps includes handling the missing values by identifying and addressing the missing data entries. Additionally, for the ECG image dataset, there are a few steps need to be taken including image resizing and flattening. The image resizing involves ensuring all images are of the same dimensions to standardize input size for models. The normalization step involves scaling pixel values to standard range (e.g., 0-1) while the data augmentation involves applying transformations like rotation, flipping and zooming the images. Lastly, after the datasets are cleaned, rigorous validation checks will be done to ensure the dataset meets predefined quality standards and fit for the design phase.

3.3 Design Phase

3.3.1 System Architecture

During the designing phase, building a system architecture remains essential because it creates a clear design to explain how prototype elements will work together to maintain efficiency along with identifying challenges during the project.

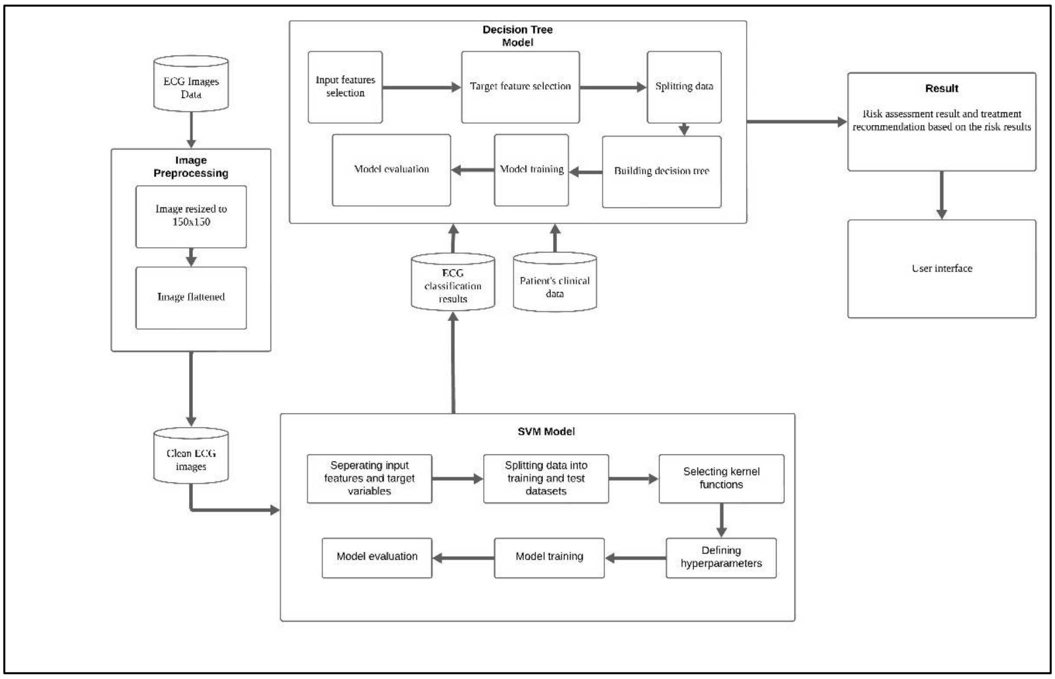

Figure 7 System architecture

Based on the **Figure 7** above, the process of the prototype begins with the raw ECG images data, which will be first redirected to the image preprocessing module. This module involves in resizing the ECG image to the standard size of 150 x 150 pixels to maintain consistency across the datasets. After the ECG image is resized, the image will be flattened into one-dimensional format resulting in clean ECG images that will be used as input features for the support vector machine (SVM) model.

Next, the clean ECG image will be processed and classified using the support vector machines (SVM) model. In this module, the model will initially separate input features from the target variables. Next, the data is split into training and test sets which to allow proper training and validation process. The next step involves selecting an appropriate kernel function such as linear, polynomial or radial basis function (RBF) for optimal classification process. Additionally, hyperparameters including regularization and gamma values will be defined to tune the model's performance. Finally, the support vector machines (SVM) will be trained on the ECG images and undergo evaluation to ensure the model is reliable and accurate. As a result, this module will produce the ECG classification result which will be used in risk assessment process using the decision tree model.

Next, the integration between the ECG classification result and the patient's clinical data will be redirected to the Decision Tree Model module. This module will assess the myocardial infarction risk level based on the input data. First of all, this model requires the input feature selection process followed by target feature selection to define the classification labels. Next, the dataset is then split into training and testing sets. The decision tree is then constructed using the training data and the model is trained. After the training process is completed, the evaluation process is conducted to measure the decision tree model accuracy and effectiveness.

Finally, the result module of this prototype is the risk assessment result and treatment recommendations based on the analysis performed by both decision tree and support vector machines models. These results will be displayed through a user interface to allow healthcare professionals to access and interpret the results.

3.3.2 Prototype Flowchart

The development of the prototype for a diagnosis and treatment recommender system for myocardial infarction using the decision tree and support vector machines (SVM) requires a flowchart as it is important to visually display system steps and algorithms integration as shown in **Figure 8**.

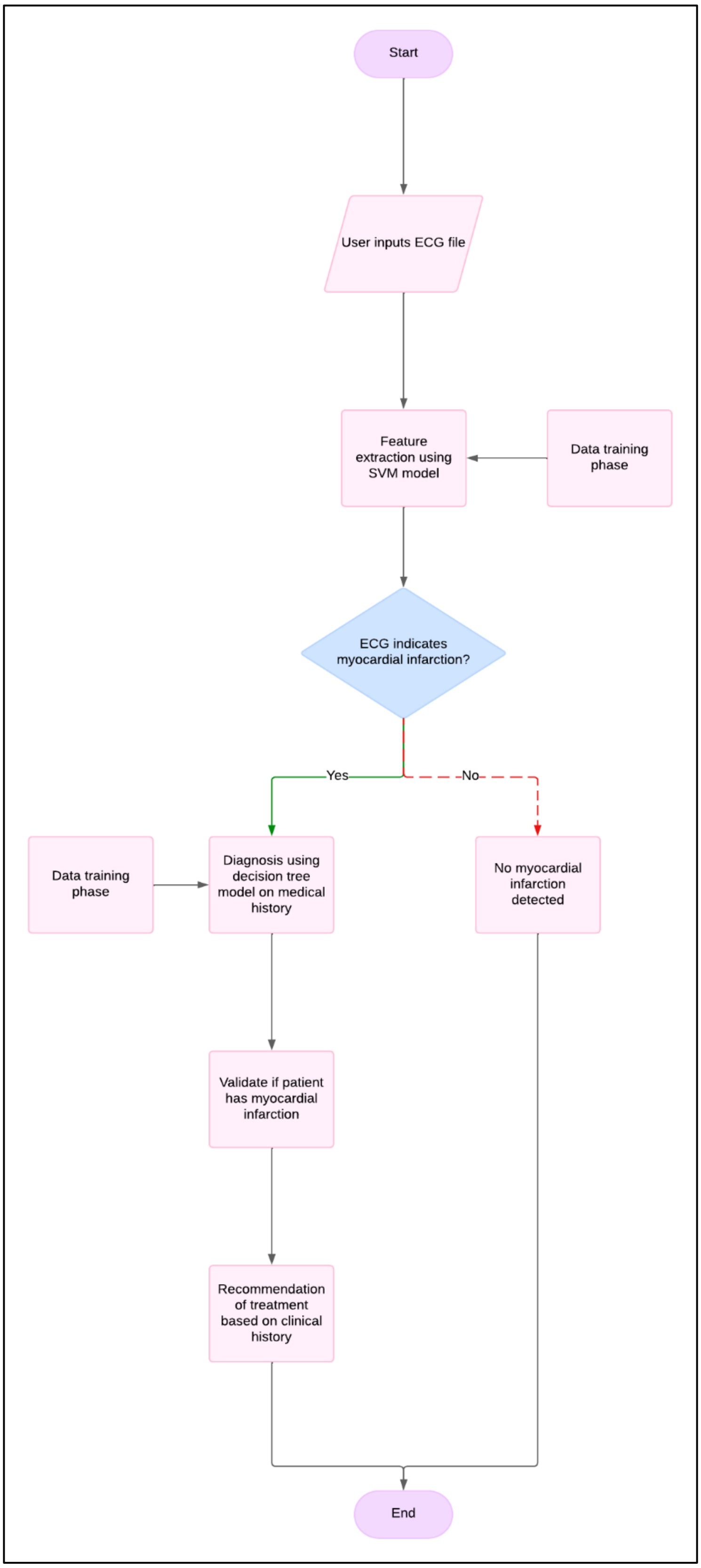

Figure 8 Prototype flowchart

Based on **Figure 8** above, the proposed prototype will start when the user uploads the patient's ECG image to the prototype. After the image is uploaded, using the Support Vector Machine (SVM), the feature extraction of the image will be done to classify whether the patient has myocardial infarction or not. If the patient's ECG image is indicated as normal, the prototype will end the process. However, if the patient's ECG image is indicated with myocardial infarction, the user must input the patient's clinical data and the diagnosis process will be done using the Decision Tree algorithm. This process will validate and classify if the patient is likely to have myocardial infarction. After that, the prototype will recommend a suitable treatment plan for the patient based on the clinical data input.

3.3.3 User Interface Design

In general, the design of the user interface of the prototype exemplifies an efficient approach for healthcare professionals. Based on the **Figure 9** below, the user interface design for this prototype consists of several key functionalities including uploading the ECG image section, patient information entry and medical history documentation. This structured layout is expected to reduce the potential for user error ultimately aiming to improve patient outcome through accurate diagnosis and personalized treatment suggestions.

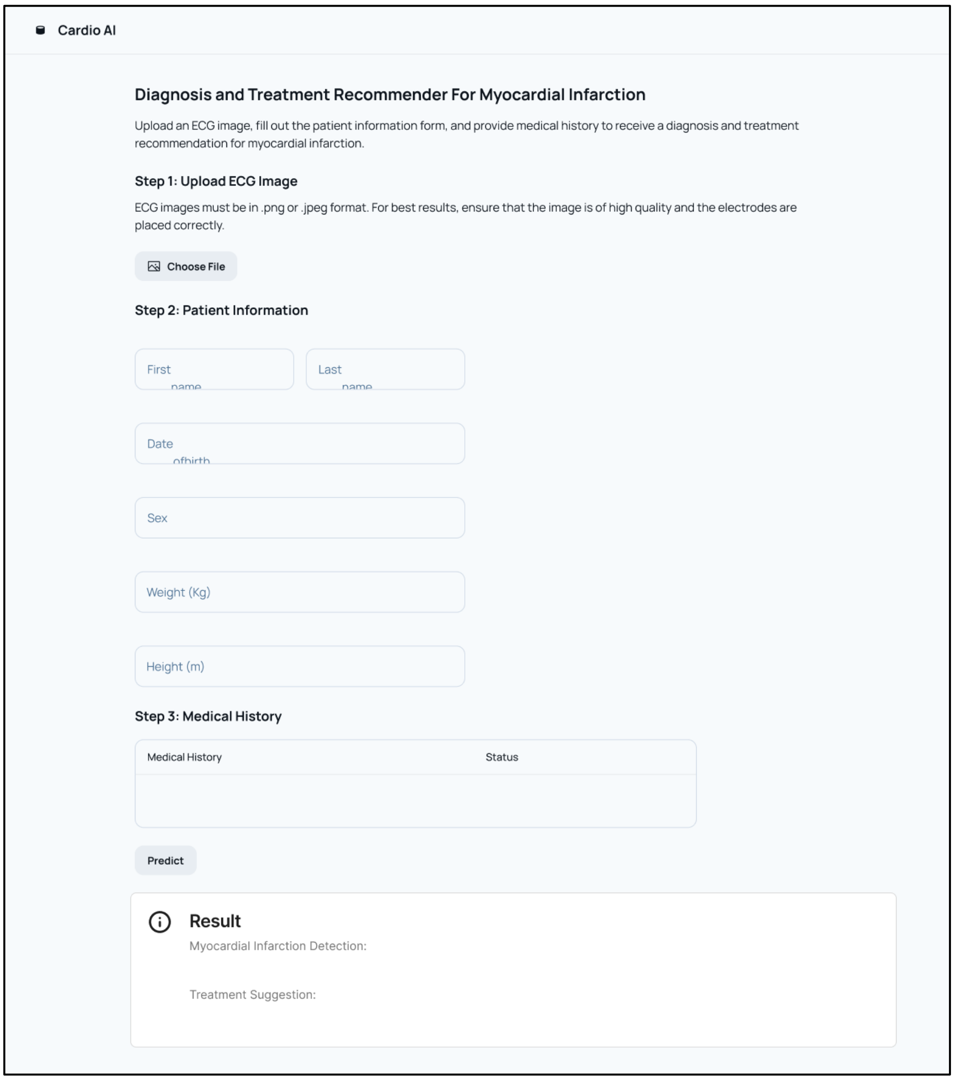

Figure 9 User interface design

This user interface begins with the first section which requires the user to upload an electrocardiogram (ECG) image. This step is crucial as the ECG image will be processed for feature extraction using Support Vector Machine (SVM) model. Following the image upload, the second section requires the user to enter the patient's information including name, date of birth, sex, weight in kilograms and height in meters. Next, the third section requires the healthcare professionals to document the patient's relevant medical history to provide additional context that may influence the diagnosis and treatment plan. Once all the necessary information is filled in, the user can click the "Predict" button which will provide the output of this prototype in the final section. In the final section, the output of the prototype will consist of myocardial infarction detection result and personalized treatment suggestions.

3.3.4 Pseudocode of Selected Algorithms

The development of a machine learning model requires pseudocode definition for the algorithm to display the structures and logics. It is essential due to its ability to visually present both algorithm structure and logics. The pseudocode create a link between human understanding and code implementation before committing to the programming languages. Hence, for this prototype, the pseudocode of decision tree and support vector machines (SVM) algorithms are constructed to visualize algorithms' structure and logic.

```
GenDecTree(Sample S, Features F)
Steps:
    1.  Ifstopping_condition(S, F) = true then
        a.  Leaf = createNode()
        b.  leafLabel = classify(s)
        c.  return leaf
    2.  root = createNode()
    3.  root.test_condition =findBestSpilt(S,F)
    4.  V = {v | v a possible outcomecfroot.test_condition}
    5.  For each value v Є V:
        a.  S_v = {s | root.test_condition(s) = v and s Є S };
        b.  Child = TreeGrowth (S_v,F );
        c.  Add child as descent of root and label the edge {root →
            child} as v
6.  return root
```

Figure 10 Pseudocode for decision tree algorithm (Hambali et al., 2019)

Based on **Figure 10** above, the pseudocode provided describes a recursive algorithm for constructing a decision tree from a dataset. The process begins by checking a stopping condition as either all samples belong to the same class or no features are left for splitting. Next, if the condition is met, a leaf node is created, classified and returned. If not, a root node is created and the best feature for splitting the data is determined. Moreover, the possible outcomes of this split are identified and for each outcome, a subset of the samples corresponding to this outcome is created. Next, the algorithm then recursively constructs subtrees for each subset attaching these subtrees to the root node and labeling the edges with the corresponding outcomes. This recursive process will continue until all stopping conditions are satisfied resulting in a full decision tree.

Training Model for SVM

Input: D=[X,Y]; X(array of input with m features), Y(array of class labels)
Y=array(C) // Class label
Output: Find the performance of the system
function train_svm(X,Y, number_of_runs)
initialize:learning_rate=Math.random();
for learning_ rate **in**number_of_runs
error=0;
for i **in** X
if (Y[i] *(X[i]*w))<1 **then**
update : w=w + learning_rate * ((X[i]*Y[i])*(-2*(1/number_of_runs)*w)
else
update: w=w+learing_rate *(-2*(1/number_of_runs)*w)
end if
end
end

Figure 11 Pseuodocode for SVM algorithm (Harimoorthy & T, 2021)

Moreover, based on the **Figure 11** above, the pseudocode for a Support Vector Machine (SVM) outlines an iterative algorithm to find the optimal hyperplane that separates different classes in the input data. The algorithm begins by initializing a random learning rate. It updates the weight vector ‘w’ for each input vector ‘x’ for a specified number of iterations based on its classification. If an input vector is misclassified or within the margin, the weight vector is adjusted by incorporating both the data point and a regularization term to minimize the hinge loss. If the input vector is correctly classified and outside the margin, then only the regularization term is used for the update. This process continues through the specified iterations, refining the weight vector to achieve the best possible classification performance.

3.3.5 Prototype Implementation

The development for this prototype will implement the waterfall model to ensure a systematic and structured approach where each phase is completed sequentially, providing clear flow and thorough documentation at every stage. First introduced by Winston W. Royce in 1970, the Waterfall Model consists of six phases: requirements phase, design phase, development phase, testing phase, deployment phase, and maintenance phase (GeeksForGeeks, 2024).

Initially, the requirement analysis phase involves the process of a preliminary study of this prototype's needs including datasets and necessary algorithms. Next, the design phase follows where the construction of detailed system architecture according to the proposed requirements. Moreover, the next phase includes the development phase which is the development of the prototype to the design specifications for the diagnostic and recommender models. In the integration and testing phase, the prototype will be tested using the test dataset and the performance of the model will be evaluated using accuracy, precision and recall metrics.

Moreover, to ensure the successful implementation the prototype, it is crucial to consider the hardware and software specifications required for the prototype as shown in **Table 5** and **Table 6** below.

Table 5 Hardware specifications

Hardware	Specifications
Processor	AMD Ryzen 5 5600H with Radeon Graphics
Memory (RAM)	8.00 GB
Graphic Memory	NVIDIA GeForce GTX 1650
Storage	Solid State Drive, 500 GB

Table 6 Software specifications

Module	Software
Front-end	Flask
Back-end	Python
Operating System	Windows 11

3.4 Performance Evaluation

3.4.1 Accuracy

One of the common metrics used to evaluate the performance of a model includes accuracy. Accuracy measures the proportion of correct predictions made by the model out of all predictions. It is calculated using the formula as shown in the **Figure 12** below.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Figure 12 Formula for accuracy

In this formula, the terms are defined as follows:

i. True Positives (TP): The instances correctly predicted as positive by the model.
ii. True Negatives (TN): The instances correctly predicted as negative by the model.
iii. False Positive (FP): The instances incorrectly predicted as positive by the model.
iv. False Negatives (FN): The instances incorrectly predicted as negative by the model.

The numerator (TP+TN) in **Figure 12** represents the total number of correct predictions made by the model while the denominator (TP+FP+TN+FN) represents the total number of instances. Hence, the accuracy evaluation is the proportion of correct predictions out of all predictions made.

3.1.1 Precision

Precision is a crucial metric in evaluation for the classification models specifically in machine learning and statistics. It is defined by a few crucial components for the calculation as shown in the **Figure 13** below.

$$Precision = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Positive(FP)}$$

Figure 13 Formula for precision (Huilgol, 2024)

The formula shown in the figure above is defined by several components:

i. True Positive (TP): The instances correctly predicted as positive by the model.

ii. False Positive (FP): The instances incorrectly predicted as positive by the model.

The high precision indicates a low rate of false positives meaning that when the model predicts a positive instance, it is likely to be correct. Conversely, low precision suggests a high rate of false positives implying that many predicted positive instances are actually negative.

3.1.2 Recall

Another essential metric used to evaluate the performance of a classification model also includes recall as shown in the **Figure 14** below.

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

Figure 14 Formula for recall (Huilgol, 2024)

For the recall metric, it consists of several components:

i. True Positive (TP): The instances correctly predicted as positive by the model.

ii. False Negative (FP): The instances incorrectly predicted as negative by the model.

The recall function measures the model's ability to identify all relevant instances within a dataset. It is the ratio of correctly predicted positive instances (TP) to the total number of actual positive instances which includes both true positives and false negatives (FN). The high recall result indicates that the model successfully identifies most of the actual positive instances while the low recall suggests that the model fails to identify many of the actual positive instances.

3.5 Gantt Chart

Constructing a Gantt chart is essential for project management as it provides a visual representation of the project's timeline. The chart outlines the start and finish dates of various tasks allowing for effective progress tracking and resource allocation. Hence, for this project, the Gantt chart constructed is as shown in **Figure 15** below.

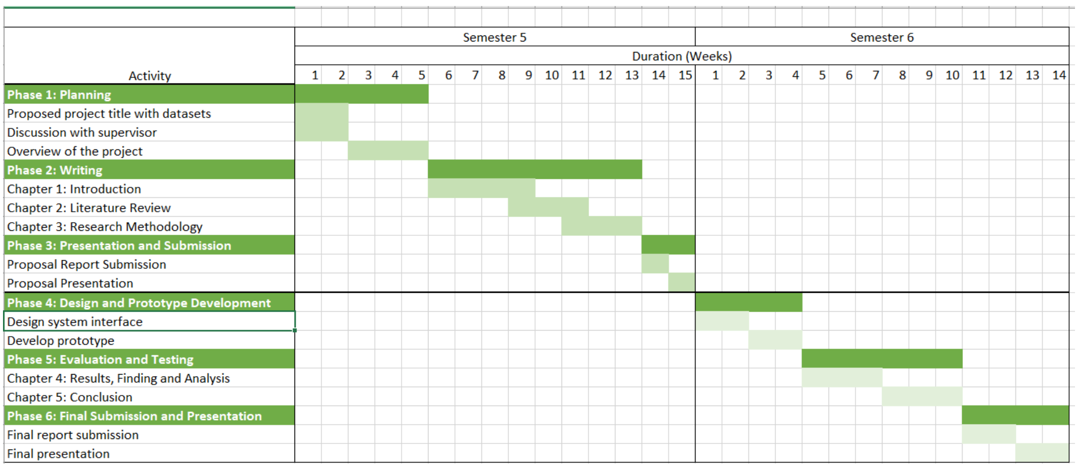

Figure 15 Project's Gantt chart

3.6 Conclusion

In conclusion, the research methodology is crucial to ensure that the proposed project is conducted systematically and objectively. A detailed research methodology helps in obtaining reliable and valid information that is reliable in theory and practical contexts. The research methodology provides a structured approach to data collection and project flow. The structured approach adopted in this research includes the preliminary study, design phase and performance evaluation provide a comprehensive framework for achieving the project objectives. Each phase is planned to address specific goals to ensure seamless progress of the development of the prototype.

CHAPTER 4

RESULT AND DISCUSSION

This chapter provides a detailed analysis of the outcomes based on the data acquired for the diagnosis and treatment recommender system for myocardial infarction using decision tree and support vector machine algorithms. This chapter also examines the outcomes generated by the algorithms, highlighting the performance and insights of the project. The diagnosis of myocardial infarction process applies both decision tree and support vector machine algorithms with both models' performance serving as an indicator of the algorithms' effectiveness in identifying the risk of myocardial infarction.

4.1 System Architecture

The system architecture is one of the most important components of the project as it defines both structural frameworks and operational sequences of the prototype, ensuring proper communication between the prototype components to achieve the desired outcomes.

Based on the **Figure 7** in System Architecture, the process of the prototype begins with the raw ECG images data, which will be first redirected to the image preprocessing module. This module involves in resizing the ECG image to the standard size of 150 x 150 pixels to maintain consistency across the datasets. After the ECG image is resized, the image will be flattened into one-dimensional format resulting in clean ECG images that will be used as input features for the support vector machine (SVM) model.

Next, the architecture implements the support vector machine (SVM) and patient's clinical processing using decision tree model to process the patient's data which includes the ECG images and clinical data to provide a diagnosis of myocardial infarction. The support vector machines (SVM) require data classification and then select a kernel before tuning hyperparameters and performing the ECG image classification. Firstly, the model will identify patterns in the ECG images to categorize them effectively. Next, a suitable kernel function such as linear of radial basis function (RBF) is chosen to enhance the classification performance. Moreover, the optimization process of this model requires setting values for the regularization parameter and kernel coefficient in order to achieve precise performance levels. Lastly, the model is trained using labelled ECG datasets to ensure it can identify abnormalities with high accuracy.

After the ECG classification process has been completed, any abnormalities result will be redirected to the risk assessment section. In this section, the patient's information including demographic and clinical data will be prompted into the prototype and this information will be processed using the decision tree model to provide the diagnosis result. Next, the system will provide the treatment plan for the patient based on the risk classification result and ended the diagnosis and treatment recommendation process.

4.2 Program Code

This segment focuses on the code implementation in the prototype to diagnose myocardial infarction using the decision tree and support vector machines algorithms and providing appropriate treatment plan for the patient.

4.2.1 Data Preprocessing

The implementation of this project required multiple essential data preprocessing steps to optimize the dataset's quality for both algorithms' model training and evaluation requirements. The distinct preprocessing was applied to the datasets for the two algorithms based on their respective data requirements.

These preprocessing tasks were fully implemented in Python, taking advantage of its powerful ecosystem which includes specific modules and packages specific for image processing and data classification process. The implementation of Python was motivated by its versatility and efficiency in handling various types of data and its support for specialized library frameworks designed for various applications.

Implementing Python programming language assist to create an optimized preprocessing process that follows the standard methods in the machine learning. A successful data preprocessing phase stands as the essential groundwork to prepare the input data for the decision tree and support vector machines (SVM) model analyses.

4.2.1.1 Support Vector Machines (SVM)

For the support vector machines (SVM), which utilizes image data, the preprocessing involved two crucial steps: image resizing and image flattening. Image refinement required both steps to achieve optimal results for classification purposes.

The initial step for the pre-processing phase for the support vector machines (SVM) involves image resizing. A uniform sizing of input data serves a crucial requirement for accurate preprocessing and classification operations by this model. This step involves adjusting the pixel dimensions of an image either enlarging or reducing it to match a specified target size.

```
img_array=imread(os.path.join(path,img))
img_resized=resize(img_array, (150,150,3)) #resize the image 150x150
```

Figure 16 Image resizing process

Figure 16 above illustrates the first step of the image resizing process which involves reading the image file using the imread function, which loads the image into a multi-dimensional array (img_array). This array represents the pixel data of the images, where each pixel's intensity of colour value is stored numerically.

After the image is loaded, the resize function is implemented to standardize the image dimensions. Based on the **Figure 16** above, the images were resized to 150x150 pixels while preserving three colour channels (RGB). This step is crucial to ensure all images have the same shape to be feed into the support vector machine algorithms. After all of the images in each categories are resized, the resized images were stored in the img_resized variable, which been used for further processing.

The next step for the pre-processing for support vector machines involves image flattening, which the process of transforming a multi-dimensional image matrix into a one-dimensional vector. This step is exclusively relevant for support vector machines (SVM) algorithm that require tabular data as input rather than multi-dimensional matrices.

```
    flat_data_arr.append(img_resized.flatten())
    target_arr.append(Categories.index(i))
print(f'loaded category: {i} successfully')
```

Figure 17 Image flattening process

Based on the **Figure 17** shown above, the flattening process flatten() method is applied to the resized image (img_resized) to transform the multi-dimensional array of pixel data into a single-dimensional array. This flattened representation retains all the pixel information while simplifying the data structure for compatibility with the support vector machine algorithms.

Next, the flattened data is appended to the list flat_data_arr, which serves as the container for all the processed image in the dataset. Simultaneously, the category label corresponding to the image is determined using categories.index(i) and appended to the target_arr list. This step ensures that each image's features are paired with its correct label, facilitating supervised learning of the algorithm.

4.2.1.2 Decision Tree

In contrast, the decision tree algorithm worked with categorical data, requiring a different preprocessing approach. The steps included dropping irrelevant columns, target feature selection, handling missing values and input feature selection. These procedures were crucial for ensuring the dataset was clean, consistent and suitable for training the decision tree model.

```
In [4]: #Drop column ID
        df.drop(label[0], axis=1, inplace=True)
        label_afterdrop=list(df.columns)

In [5]: input_label = label_afterdrop[0:111]
        print(f'Number of features after dropping irrelevant column: {len(label_afterdrop)}')
        print(f'Number of features input label: {len(input_label)}')

        Number of features after dropping irrelevant column: 123
        Number of features input label: 111
```

Figure 18 Dropping irrelevant columns

Based on the **Figure 18** above, the process involves dropping an irrelevant column which is the column that represents the patient's ID. Using the drop() method, the label[0] column is removed from the dataset. The axis=1 parameter was applied to ensure that the column removal operation only targets a column while the inplace=True argument was applied to run the changes directly to the original DataFrame.

After the patient's ID was dropped, the remaining feature names are stored in a list called label_afterdrop. Furthermore, a subset of these features is selected by defining the list to extract the first 111 columns as the input_label variable. The next data preprocessing step for the decision tree model involves the target feature selection, which is the process of choosing the feature that represents the outcome or dependent variable for the model to predict.

```
In [6]: #Target feature
        target_feature=label[118]
        print(f'Target features that cause death: {target_feature}')

        Target features that cause death: RAZRIV
```

Figure 19 Target feature selection

Based on the **Figure 19** above, the selection process begins by identifying the appropriate column in the dataset from the range of columns 112 to 123. For this project, the feature label[118], which is also the myocardial rupture column was chosen for the desired outcome for the prediction. Hence the column was assigned to the variable target_feature and the validation process was applied, in which a print() statement was included. The output of the name of the chosen target feature "RAZRIV" validated that the correct feature was selected.

After the target feature was selected, the next step for data preprocessing was taken which involved handling missing values. This process is fundamental as missing data can significantly impact the performance and accuracy of the model.

```
In [7]: #Check for number of missing value
        missing_value = df.isnull().sum()
        print(f'Total missing value: {sum(missing_value)}')

        Total missing value: 15974
```

Figure 20 Handling missing value

Based on **Figure 20** above, the process of handling missing values begins by checking for the number of missing data. The dataset's missing values were identified using the isnull() function from the Pandas library which detects null or NaN entries across all columns. By using the sum() method, the total number of missing values was calculated and display the total of 15,1974 missing data highlighting the extent of incomplete data within the dataset.

```
In [8]: #fill missing value with column mean
        df = df.fillna(df.mean())

        print(df)

              AGE  SEX  INF_ANAM  STENOK_AN  FK_STENOK  IBS_POST  IBS_NASL   GB  \
        0     77.0    1       2.0   1.000000   1.000000  2.000000     0.375  3.0
        1     55.0    1       1.0   0.000000   0.000000  0.000000     0.000  0.0
        2     52.0    1       0.0   0.000000   0.000000  2.000000     0.375  2.0
        3     68.0    0       0.0   0.000000   0.000000  2.000000     0.375  2.0
        4     60.0    1       0.0   0.000000   0.000000  2.000000     0.375  3.0
        ...    ...  ...       ...        ...        ...       ...       ...  ...
        1695  77.0    0       0.0   4.000000   2.000000  1.000000     0.375  2.0
        1696  70.0    0       0.0   6.000000   2.000000  1.000000     0.375  2.0
        1697  55.0    1       3.0   6.000000   2.000000  2.000000     0.375  0.0
        1698  79.0    0       2.0   2.000000   2.000000  1.000000     0.375  2.0
        1699  63.0    1       2.0   2.316186   1.205286  1.160703     0.375  2.0
```

Figure 21 Filling the missing values

After the number of missing values was identified, the missing entries were filled with the column mean as shown in **Figure 21**. This strategy was implemented using the fillna() method from the Pandas library which by replacing all the null or Nan values with the mean for each columns. Then, the dataset was updated with these imputed values to ensure that all missing data was addressed.

```
In [9]: #result

        missing_value = df.isnull().sum()
        print(f'Total missing value: {sum(missing_value)}')

        Total missing value: 0
```

Figure 22 Validate for missing value handling

After all of the missing data is addressed, the validation process will take place by using the isnull() function from the Pandas library. The total number of missing values displayed was 0 as shown in **Figure 22**, indicating that the handling of the missing value process has been success.

```
In [11]: #Define the selected features
         input_label=[
             'AGE','SEX','INF_ANAM','STENOK_AN','FK_STENOK','GB','ZSN_A',
             'nr_11','endocr_01','endocr_02','S_AD_KBRIG','D_AD_KBRIG',
             'ant_im','lat_im','inf_im','post_im']

         print(f'Selected features: {input_label}')

         Selected features: ['AGE', 'SEX', 'INF_ANAM', 'STENOK_AN', 'FK_STENOK', 'GB', 'ZSN_A', 'nr_11', 'endocr_01', 'endocr_02', 'S_AD
         _KBRIG', 'D_AD_KBRIG', 'ant_im', 'lat_im', 'inf_im', 'post_im']
```

Figure 23 Input features selection

After all of the missing values were addressed and successfully handled, the final step of the data preprocessing for the decision tree model requires the defining of selected input features process. **Figure 23** above shows the code to define a selected feature list into a variable called input_label. These features were chosen from the dataset because of their relevance in diagnosing myocardial infarction. The list includes features such as demographic information including age (AGE), gender (SEX), diabetes (endocr_01), obesity (endocr_02) and clinical information such as previous myocardial infarctions history (inf_anam), arrhythmia history (nr_11), angina pectoris history (STENOK_AN), functional class of angina (FK_STENOK), hypertension stage (GB), heart failure stage (ZSN_A), systolic (s_adkbrig) and diastolic (d_ad_kbrig) blood pressure. Moreover, the ECG features were also included such as anterior (ant_im), lateral (lat_im), inferior (inf_im) and posterior (post_im) myocardial infarctions. These features next are printed using the print function to verify the input_label variable were assigned with the correct list of features.

4.2.2 Implementation Of Algorithms

The implementation of algorithms is the most critical aspect of this project as it forms the base of the prototype's operations and determines its overall performance. For this project, the support vector machines and decision tree algorithms were integrated to address the project's objectives.

4.2.2.1 Support Vector Machines (SVM)

The implementation of support vector machine begins by creating a structured pandas DataFrame to organize data for the support vector machine analysis. A DataFrame is a two-dimensional, tabular data structure commonly used in Python for handling and processing datasets.

```
In [4]: #dataframe
        df=pd.DataFrame(flat_data)
        df['Target']=target
        df.shape

Out[4]: (928, 67501)
```

Figure 24 Creating DataFrame

The construction of DataFrame begins by assigning pd.DataFrame(flat_data), where flat_data is assumed to be a multidimensional array or matrix containing the features, into the df variable. Next, the Target column is added to the DataFrame using df['Target'] = target which in this part, the Target column held the target variable, which is typically the output or class labels for a classification task or the dependent variable for a regression task. Finally, the df.shape function showed the number of rows (observations) and columns (features plus the target variable) which outputs 928 observations and 67,501 columns including the target variable as shown in **Figure 24**.

Next, the implementation of support vector machine requires the process of separating input features and target variables from the dataset, which is a critical step in preparing the data for support vector machine model.

```
In [5]: #Separate input features and targets

        #input data
        x=df.iloc[:,:-1]
        #output data
        y=df.iloc[:,-1]
```

Figure 25 Separating input features and target variables

Figure 25 above shows the input features (x) which serve as predictors while the target variable (y) represents the output that the model aims to predict. Using iloc, the input features are selected by slicing all rows (:) and all columns except the last column (:-1) as assuming the last column contains the target variable.

```
In [6]: #Separate input features and targets

        #Splitting the data into training and testing sets
        x_train,x_test,y_train,y_test=train_test_split(x,y,
                                                       test_size=0.20,
                                                       random_state=77,
                                                      stratify=y)
```

Figure 26 Splitting data into training and testing sets

After the input features and target variables were separated, the dataset is prepared for training evaluation by splitting it into training and testing subsets. The process is implemented using the train_test_split function. **Figure 26** above shows the test_size parameter at 0.20 splits the data 80:20 for training and testing. The random_state parameter was assigned a value of 77 to guarantee reproducibility. Additionally, the stratify parameter is specified with the target variable y to preserve the target variable class distribution between training and testing sets.

```
In [7]: #Defining the parameters grid for GridSearchCV
        param_grid={'C':[0.1,1,10,100],
                    'gamma':[0.0001,0.001,0.1,1],
                    'kernel':['rbf','poly']}

        #Creating a support vector classifier
        svc=svm.SVC(probability=True)

        #Creating a model using GridSearchCV with the parameters grid
        model=GridSearchCV(svc,param_grid)
```

Figure 27 Tuning of hyperparameters

Furthermore, implementing the support vector machine (SVM) requires careful tuning of hyperparameters to achieve maximum output performance. Based on the **Figure 27**, the hyperparameter tuning is performed using GridSearchCV which involves defining the parameter grid, param_grid with three key hyperparameters: C (Regularization Parameter), Gamma (Kernel Coefficient) and Kernel. In this code, the C parameter values are defined with values [0.1, 1, 10, 100] to determine the optimal regularization strength. Next, the Gamma (Kernel Coefficient) is set to values [0.0001, 0.001, 0.1, 1] as the smaller values of gamma result in a smoother decision boundary. Finally, the two types of kernels tested in this code are the Radial Basis Function (RBF) and polynomial kernels.

Next, an SVM model is created with the svc=svm.SVM(probability=True) statement to enable the output of class probabilities. The GridSearchCV function is used to integrate the SVM model with the defined parameter to evaluate each hyperparameter combination.

4.2.2.2 Decision Tree

Similarly, the implementation of decision tree begins by creating a structured pandas DataFrame to organize data for analysis process. A DataFrame is a two-dimensional, tabular data structure commonly used in Python for handling and processing datasets.

```
#Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 28 Splitting data

The dataset is prepared for training evaluation by splitting it into training and testing subsets using the train_test_split function. According to **Figure 28** above, the test_size parameter is divided at 20% for testing with the rest allocated for training process. The random_state parameter is assigned a value of 42 to ensure each execution produces the same results. Moreover, the stratify parameter is specified with the target variable y to ensure that the class distribution in the target variable is maintained across both the training and testing subsets.

```
In [68]: from sklearn.utils.class_weight import compute_class_weight
         import numpy as np

         # Calculate class weights based on dataset imbalance
         classes = np.unique(y_train)
         class_weights = compute_class_weight(class_weight='balanced', classes=classes, y=y_train)

In [69]: # Convert to dictionary
         class_weight_dict = dict(zip(classes, class_weights))
         print("Calculated Class Weights:", class_weight_dict)

         Calculated Class Weights: {0: 0.5163249810174639, 1: 15.813953488372093}
```

Figure 29 Class weighting

Next, the implementation of a decision tree model involves addressing the challenge of class imbalance in the dataset as shown in **Figure 29**. Class imbalance in this dataset occurs as the class 0 (Low risk of myocardial infarction) is significantly more than the class 1 (High risk of myocardial infarction). Hence, the compute_class_weight function is utilized to calculate the weights for each class. The parameter class_weight= 'balanced' indicates the function will automatically compute weights inversely proportional to the class frequencies in the training dataset (y_train). The calculated weights are then converted into a dictionary using the zip function, mapping each unique class label to its corresponding weight. As the result, in the output, class 0 is given a smaller weight of approximately 0.5163, while class 1 is assigned a much higher weight of 15.8139. This distribution ensured that the decision tree model will consider the minority class more carefully during the training phase.

```
In [70]: # Using calculated class weights
         dt_model = DecisionTreeClassifier(random_state=42, class_weight=class_weight_dict)
         dt_model.fit(X_train, y_train)
         y_pred_dt = dt_model.predict(X_test)
```

Figure 30 *Data training*

Furthermore, to utilize the calculated class weights effectively, the DecisionTreeClassifier from the sklearn.tree module is implemented as shown in

. The decision tree classifier is instantiated with two critical parameters: random_state and class_weight. Next, the fit method is used to train the decision tree classifier on the training dataset (X_train and y_train). By incorporating the class weights, the model adapts its learning process to give higher priority to underrepresented classes, reducing the risk of bias. Once the model is trained, predictions are generated on the test dataset (X_test) using the predict method.

4.3 User Interface

One of the most important parts of this project is the graphical user interface which provides the user an interactive platform to ensure ease of navigation and efficient access in diagnosing myocardial infarction and recommending treatment plans to the patients. For this project, the user interface was developed using the Flask tools and divided into certain parts.

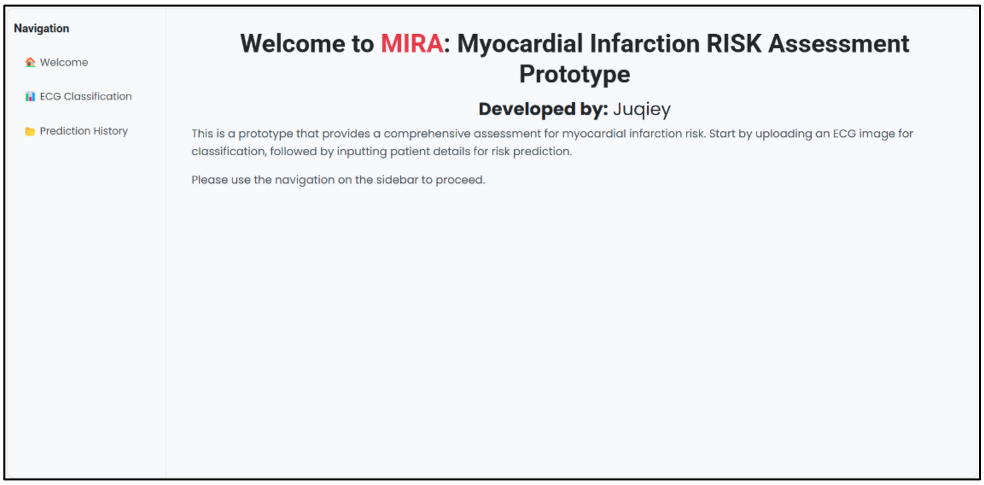

Figure 31 Welcome page

The first part of the user interface for this project is the welcome page as shown in the **Figure 31** above. In this page, the navigation menu on the left helps users to explore the prototype's features. To access the diagnosis and treatment recommendations page, the user must select the ECG Classification button to access the page.

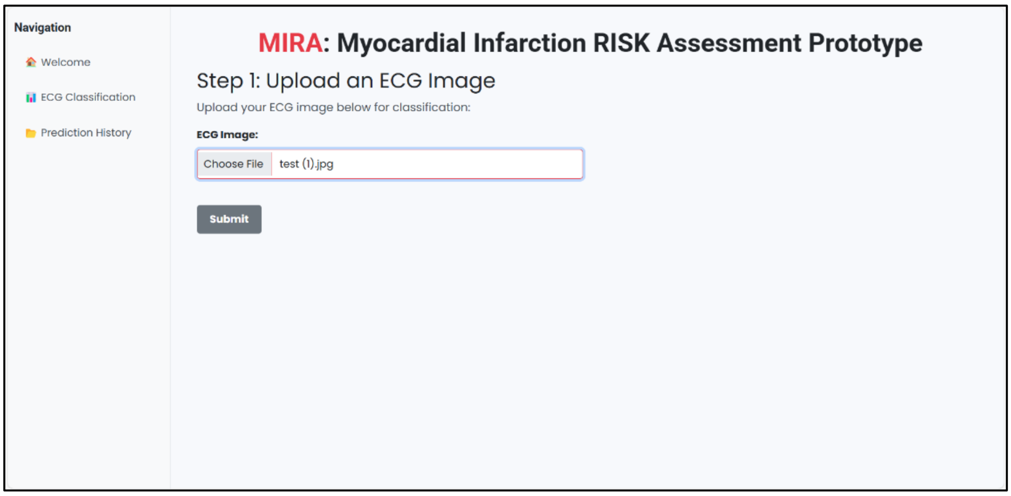

Figure 32 ECG classification section

After the user is redirected to the ECG classification page, the first section of the diagnosis and treatment page is the ECG classification section as in

. In the ECG classification section, the user is required to upload the ECG image in JPG, JPEG or PNG format. After the image is uploaded, the image will be processed and classified using the SVM model.

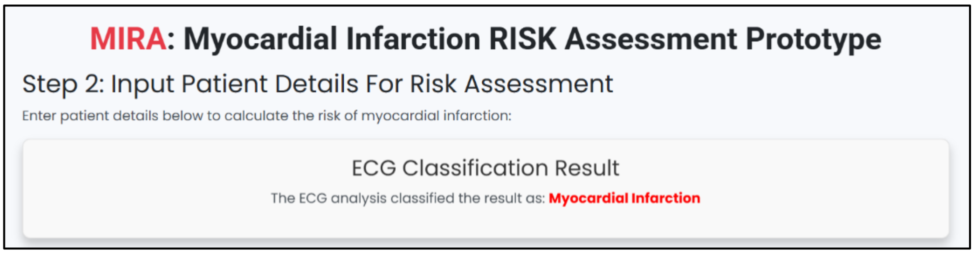

Figure 33 Abnormal ECG result page

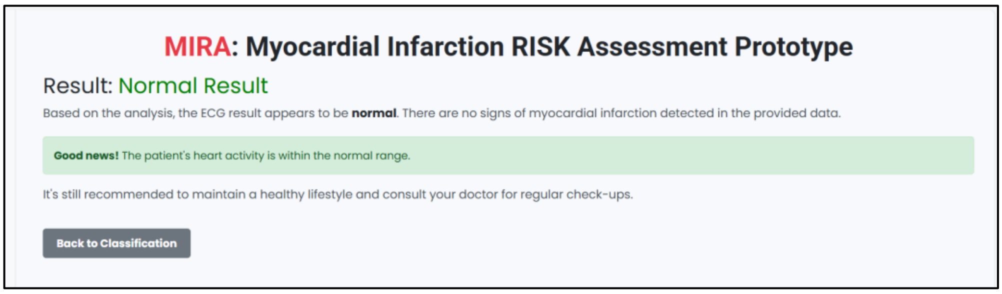

Figure 34 Normal ECG result page

After the classification process is finished, the prototype will display the predicted ECG classification results as shown in **Figure 33** and **Figure 34**. If the ECG shows any abnormalities as shown in **Figure 33** such as myocardial infarction, history of myocardial infarction or abnormal heartbeat, the user will be redirected to the risk assessment page. However, if the ECG classification result is normal, the user does not need to continue to the risk assessment section and the process will stop.

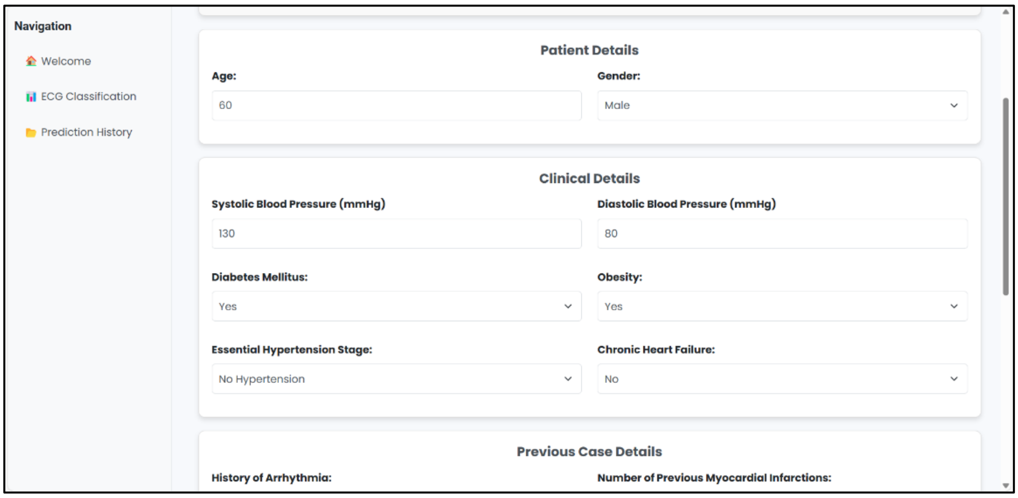

Figure 35 Risk assessment section

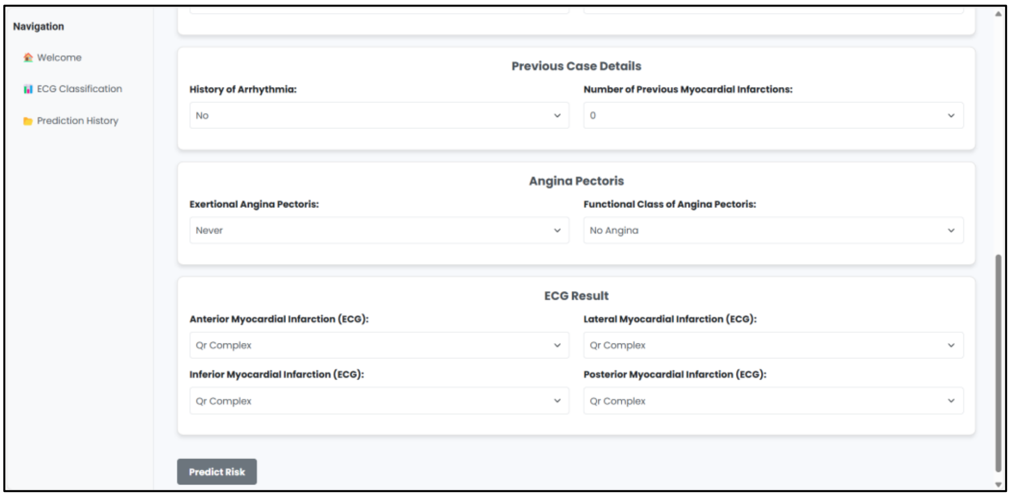

Figure 36 Continuation of risk assessment page

Based on the **Figure 35** and **Figure 36** above, the risk assessment section requires the user to input patient details. User can specify the patient's age and gender along with a range of clinical details such as number of previous myocardial infarctions etc. This step ensures the prototype considers the relevant medical factors to provide a precise risk evaluation. After all inputs are filled, the user is required to push the predict button to begin the prediction process using decision tree model.

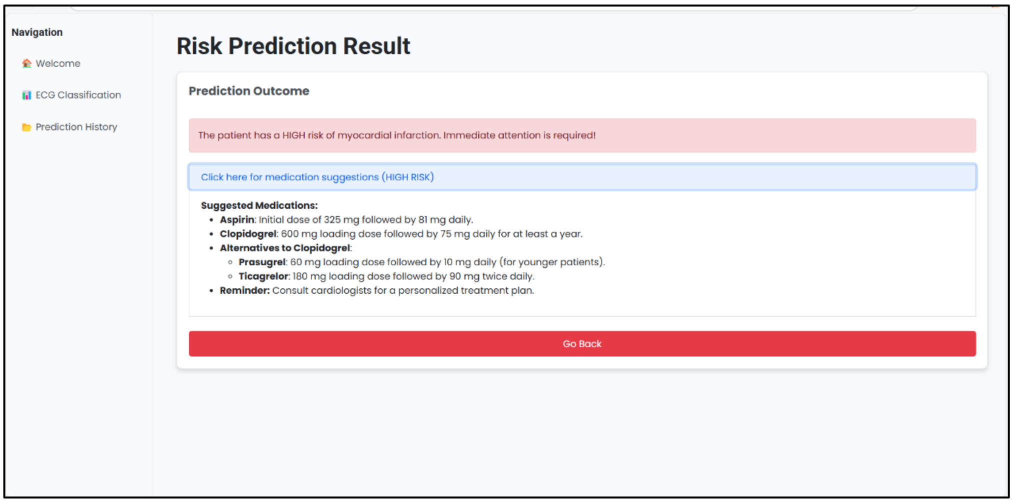

Figure 37 Prediction and treatment result

After the prediction process is finished, the prototype will display the risk percentage result for the patient and suggest the medication treatment as shown in **Figure 37**.

4.4 Evaluation Result

Models' evaluation is a process that involves assessing the performance of a machine learning model to determine its effectiveness and reliability especially in the medical diagnosis process.

4.4.1 Support Vector Machines (SVM)

For this project, the performance of the support vector machine (SVM) model is measured using accuracy, precision, recall, F1-score and confusion matrix. For the SVM model, the most common practice in evaluating the model is based on data splitting which to ensure the performance is tested on both training and unseen data. **Table 7** provides the performance of the support vector machine (SVM) for different splits including 70-30, 80-20, and 90-10 using the poly kernel.

Table 7 SVM performance

Number of Images	Kernel	Split	Accuracy	Precision	Recall	F1-Score
928	Poly	70-30	92.47%	91.89%	92.25%	92.05%
		80-20	94.62%	94.46%	94.34%	94.38%
		90-10	94.62%	95.01%	94.20%	94.52%

Based on the analysis of the SVM performance, the most optimal data split to achieve the maximum performance is the 80-20 split with an accuracy of 94.62%. Even though the 90-10 split managed to achieve better performance of precision with 95.01% compared to 80-20 with the precision of 94.46%, the split managed to outperform the recall result of 94.34% which the recall performance is very important for the medical diagnosis (Kumar, 2024). This analysis suggests that the 80-20 split is the most optimal split, advanced the model performance in ECG classification.

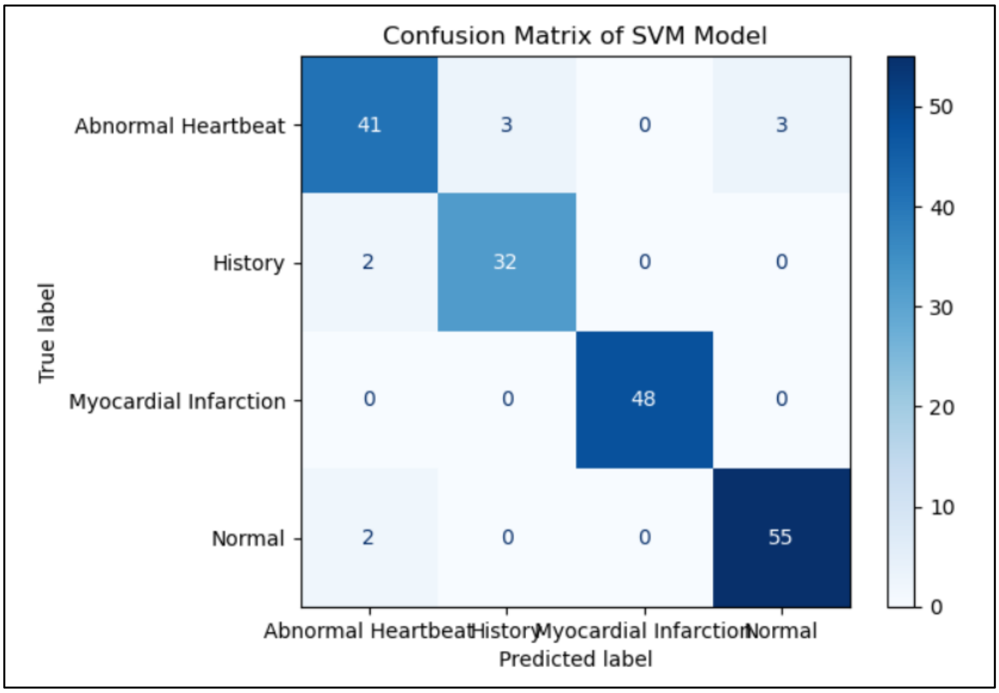

Figure 38 SVM confusion matrix

Moreover, the confusion matrix in **Figure 38** above for the 80-20 split illustrates the performance of the SVM model across four classes: Abnormal Heartbeat, History of Myocardial Infarction, Myocardial Infarction and Normal. Overall, the model performed well as seen in the high number of true positives for each class. For example, 41 instances of Abnormal Heartbeat were correctly classified, alongside 32 for History of Myocardial Infarction, 48 for Myocardial Infarction and 55 for Normal. However, some misclassifications occurred especially in the Abnormal Heartbeat category where 6 instances were incorrectly predicted. The History class on the other hand only misclassified 2 instances as Abnormal Heartbeat. On the contrary, the model achieved perfect classification for the Myocardial Infarction class with no false negatives or false positives.

4.4.2 Decision Tree

In the other hand, the performance of the decision tree model is also summarized using various metrics that assess its predictive capabilities including accuracy, precision, recall and F1-score.

The performance of the decision tree model is summarized using various metrics that assess its predictive capabilities including accuracy, precision, recall and F1-score. **Table 8** provides the performance of the decision tree for different splits including 70-30, 80-20, and 90-10.

Table 8 Decision Tree performance

Number of data	Split	Accuracy	Precision	Recall	F1-Score
1700	70-30	94.90%	94.58%	94.90%	94.74%
	80-20	96.47%	96.47%	96.47%	96.47%
	90-10	94.71%	94.25%	94.71	94.47%

Based on the analysis of the decision tree performance, the most optimal data split to achieve the maximum performance is the 80-20 split with an accuracy of 94.62%. In comparison, the accuracy performance for 80-20 is the highest compared to 70-30 and 90-10 splits which achieved the accuracy of 94.90% and 94.71% respectively. Besides, the performance for the precision, recall and F1-Score for the 80-20 are also the highest at the value of 96.47%.

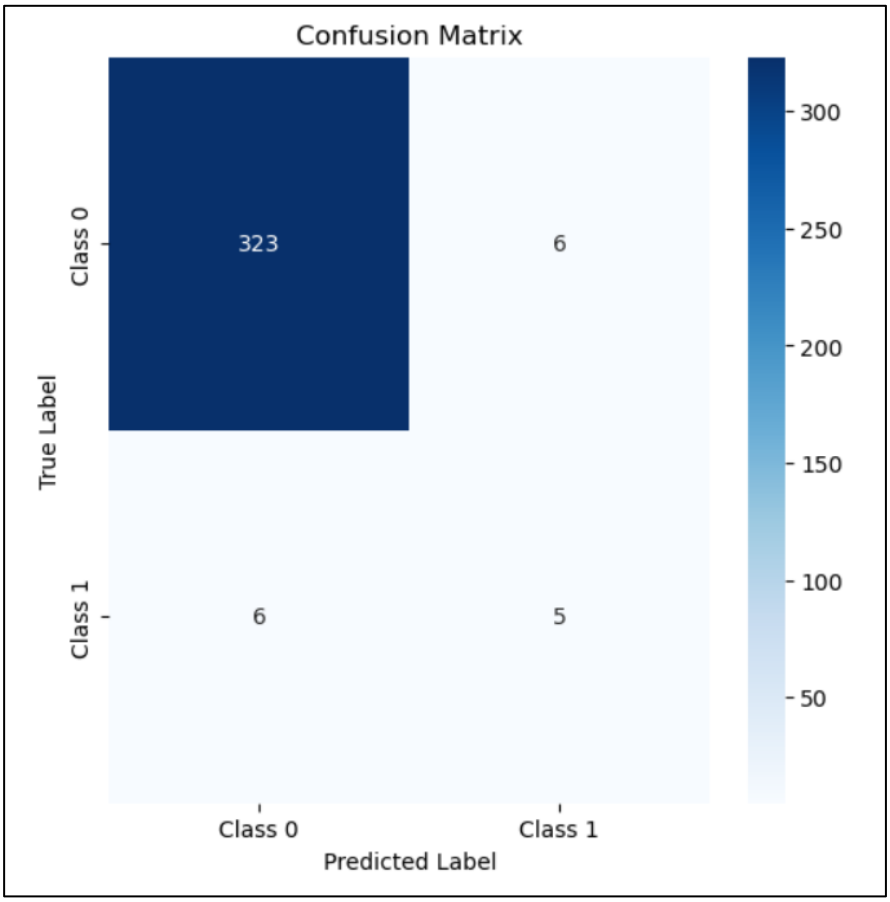

Figure 39 Decision Tree confusion matrix

Next, the decision tree performance is also measured using the confusion matrix. Based on the **Figure 39**, in the top-left cell, the value 323 represents the number of true negatives. These are instances where the actual label was class 0 and the model correctly predicted them as class 0. This high number indicates that the model is highly effective at identifying and correctly classifying the majority class. In the top-right cell, the value 6 represents the number of false positives. These are cases where the actual label was class 0 but the model mistakenly predicted them as class 1. These misclassifications could occur because the model struggles with clear boundary definitions for the minority class due to the class imbalance in the dataset.

In the bottom-left cell, the value 6 represents false negatives. These are instances where the actual label was class 1 but the model incorrectly classified them as class 0. False negatives are particularly critical in situations where missing a positive case can have significant consequences. In the bottom-right cell, the value indicates true positives. These are cases where the actual label was class 1 and the model correctly predicted them as class 1.

4.5 Discussion

The evaluation of the diagnosis and treatment recommender system for myocardial infarction demonstrates an accuracy of 96.7% for the decision tree and 94.62% for the Support Vector Machines. To provide context for these findings, the results are compared with similar works in the field.

For instance, Rambe & Mandala (2023) conducted a research study titled "Myocardial Infarction Detection as Element of Preventive Medicine with Random Forest" which attained an accuracy of 89.13%. Moreover, Liu et al. (2023) also undertook research titled "Prediction Model of Heart Attack Based on Optimized Decision Tree Algorithm" and obtained an accuracy of 80.23%. Ibrahim et al. (2023) also conducted research entitled "Heart disease Prediction using Machine Learning" in which the decision tree managed to achieve an accuracy of 78%. Hence, it shows the significant advancement of the decision tree model in this project which outperformed most of the similar model performance.

On the other hand, the support vector machine also managed to achieve significant performance compared to other similar works. For instance, Sharma (2023) conducted a research study titled "Prediction and Analysis of Heart Attack using Various Machine Learning Algorithms" and achieved an accuracy of 91.85% using support vector machine (SVM). Moreover, Bera (2024) in his research study entitled “Deep learning approach for MI detection using SVM-based classifier” attained an accuracy of 97.37%. Mahin & Ahmad (2021) also conducted a research study entitled “Diagnosis of Left Ventricular Hypertrophy from ECG Signals Based on CCS Methodology using SVM” also managed to achieve the accuracy of 80%. This also the indicate the notable performance of support vector machine model for this project compared to other similar works.

However, during the development of the algorithms model, there were a few major weaknesses that needed to be addressed. For instance, the major weakness of the decision tree is the tendency of poor performance with class-imbalanced datasets. This weakness tends to make decision tree prioritize the dominant class which in this project, the non-myocardial infarction class which leads to misclassification of the minority class, the myocardial infarction class. Hence to overcome this weakness, the class weight-adjusting technique was implemented to give higher importance to the minority class during training.

4.6 Conclusion

In conclusion, this chapter outlines the system architecture, illustrating workflow from data preprocessing to user input and output. It also provides a detailed explanation of the program codes, demonstrating the functionality of the decision tree and support vector machine. Furthermore, the chapter discusses the evaluation process including the techniques utilized and the results obtained. **Table 9** below is the summary of this project including the background, problem statement, objectives, scope, literature review, phases, activities conducted, methodology and project outcomes.

Table 9 Project Summary

DIAGNOSIS AND TREATMENT RECOMMENDER FOR MYOCARDIAL INFARCTION USING DECISION TREE AND SUPPORT VECTOR MACHINES (SVM)								
BACKGROUND	PROBLEM STATEMENT	OBJECTIVE	SCOPE	LITERATURE REVIEW	PHASE	ACTIVITY	METHOD	DELIVERABLE OUTPUT
Myocardial Infarction: **A condition caused by blocked blood flow to the heart, leading to tissue damage and potential life-threatening complications.** **Diagnostic Methods:** **Electrocardiograms (ECG) and biomarkers, such as troponins, are essential tools for**	**Diagnosis Challenges Despite Technology:** Despite the significant advances in the implementation of technology in healthcare, accurately diagnosing diseases remains a global challenge due to the complexity of the various disease	To investigate the implementation of the decision tree and support vector machines (SVM) algorithm for medical diagnosis and treatment plan recommendation for myocardial infarction patients.	**User:** Healthcare professionals involved in the diagnosis and treatment of myocardial infarction including cardiologists, emergency room physicians and clinical staff. **Datasets:** Two datasets from Kaggle	Implementation of Decision Tree and Support Vector Machines in diagnosing myocardial infarction.	**Preliminary phase** 1.Preliminary study of decision tree and support vector machines (SVM). 2.Knowledge acquisition of decision tree and support vector machines. 3.Data pre-processing	Literature Review Data Pre-processing	Reading journals, articles, book and forums. Watching videos from Youtube on related topics and techniques. Data collection Data cleaning	Knowledge Decision Tree and Support Vector Machines (SVM) performance on diagnosing myocardial infarction. Research objectives and scopes Testing and evaluation methods used

accurate and timely diagnosis. **Treatment: Management includes prehospital care, pharmacologic interventions to stabilize the patient, reperfusion therapies like angioplasty to restore blood flow, and rehabilitation to aid recovery and prevent recurrence.** **Machine Learning In Diagnosis: Advanced algorithms such as decision trees and**	mechanisms and the underlying symptoms (Alowais et al., 2023).		will be used in the prototype: one comprising 1,700 entries across 124 demographic columns and another consisting of 928 ECG images with 4 classifications. **Process:** 1.Collecting and pre-processing data from Kaggle. 2.Training models and evaluating					Clean dataset acquisition
	Negligence in Detecting Myocardial Infarction: According to the Mantle (2021), cardiologists often face negligence claims for failing to diagnose or	To develop a prototype that will provide accurate diagnosis and treatment plan for myocardial infarction using decision tree and SVM.		1.Similar works 2.Implications of literature review	**Design and Implementation Phase** 1.Design phase 2.Development phase	System design and Algorithm development	Define system requirements and specifications Develop the user interface and other system components	Detailed Design Document The system architecture of prototype. Flowchart of the prototype Model training.

support vector machines (SVM) enhance diagnostic accuracy by analyzing complex medical data patterns).	treat myocardial infarction properly with up to 25% of cardiac events going unrecognized. The common reasons for these claims include not suspecting myocardial infarction and misinterpretation of ECG results.2023).		algorithm performance. 3. Creating a user-friendly interface for healthcare professionals. **Algorithm:** Decision Tree & Support Vector Machines (SVM) Mobile Technology:					Graphical user interface of prototype. Diagnosis and treatment plan recommender for myocardial infarction using Decision Tree and Support Vector Machine (SVM) prototype,
	Population Growth Straining Healthcare: The rapid growth of population caused the	To evaluate the performance of the decision tree and SVM algorithms in making diagnosis and			**Evaluation phase** 1.Model evaluation methods 2. Testing	Performance evaluation	Use a test dataset to evaluate prototype performance. Calculate	**Decision Tree** Accuracy: 96.47% Recall: 96.47% Precision: 96.47% F1-score: 96.47%

	diagnosis and treatment planning process become difficult (Bharath i et al., 2023)	treatment plan for myocardial infarction patients.		metrics such as accuracy, f1-score, precision and recall.	**Support Vector Machines (SVM)** Accuracy: 94.62% Recall: 94.34% Precision: 94.46% F1-score: 94.48%

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

This chapter presents a detailed project summary by covering the development process and progress achieved. The key components highlighted in this project is not only the project successes but also the challenges encountered during the development. Acknowledging these limitations provides transparency which helps the readers to understand the context of the diagnosis and treatment recommender system using decision tree and support vector machines algorithms. Moreover, this chapter also provides meaningful insights for the improvement of the developed prototype in the future. In summary, this chapter reflects on the whole project from the beginning to the end.

5.1 Project Summary

The first objective of this project is to investigate the implementation of the decision tree and support vector machine (SVM) algorithms for medical diagnosis and treatment plan recommendation for myocardial infarction patients. From this project, the knowledge of the operational mechanism of decision tree and support vector machine has been successfully acquired. Moreover, the evaluation metrics used to measure the performance of these algorithms have also been successfully defined.

The second objective of this project is to develop a prototype that will provide accurate diagnosis and treatment plans for myocardial patients based on the patient's medical history using the decision tree and support vector machines (SVM) algorithms. The support vector machine (SVM) algorithm was used to classify the ECG images into four categories: normal, myocardial infarction, abnormal heartbeat and history of myocardial infarction. Next, based on the

ECG classification results and the patient's medical history, the decision tree algorithm was implemented to classify myocardial infarction risk level and provide the treatment plan recommendation.

The third objective is to evaluate the performance of the decision tree and support vector machines (SVM) in diagnosing and recommending treatment plans for myocardial infarction patients. The evaluation metrics used to evaluate the algorithms' performance in this project is accuracy, precision, recall and F1-Score. From the evaluation results, the decision tree managed to achieve an accuracy of 96.47%, 96.47% in precision, 96.47% in recall and 96.47% in F1-score. For the support vector machine (SVM), this algorithm managed to achieve an accuracy of 94.62%, 95.01% in precision, 94.34% in recall and 94.38% in F1-score.

5.2 Project Contribution

This project makes notable contributions to both the technological and medical field. From a technological insight, this project demonstrates the successful application of decision tree and support vector machines in the myocardial infarction diagnosis and treatment plan prototype. By implementing these machine learning algorithms, the project managed to demonstrate the performance of the algorithms in a medical context, particularly in dealing with complex and high-dimensional datasets.

From a medical perspective, the project contributes to improve diagnosing and recommending treatment plans for myocardial infarction patients. The developed prototype can identify myocardial infarction cases and also provides appropriate treatment recommendations based on the patient's condition.

5.3 Project Limitation

The limitation of this project is the unavailability of the latest medical dataset especially in the local context. The datasets used for this project were obtained from public repositories and lacked sufficient representation of demographic and clinical variations. Hence, this limitation may cause the prototype is not applicable in real-world scenarios.

Another significant limitation in this project is the issue of imbalanced datasets. This imbalance issue can lead to biased model performance where the algorithms become overly performed at predicting the majority classes while underperforming on minority classes.

5.4 Project Recommendation

For future recommendations and to address the limitations of this project, collaboration with local hospitals or medical research institutions can be done to acquire an up-to-date and more relevant dataset. Datasets that consist of local demographics and the latest disease patterns will significantly improve the performance of this prototype.

Moreover, while this project focuses on myocardial infarction, future developments could widen the prototype's application to other medical conditions. Utilizing multi-modal data such as imaging, lab results and others can improve the prototype performance across medical scenarios. Additionally, incorporating features such as real-time alerts and mobile compatibility could make the prototype more practical use.

While the current prototype utilizes the support vector machine (SVM) and decision tree algorithms, exploring any additional machine learning algorithms such as Convolutional Neural Network (CNN) for ECG image analysis may improve the prototype performance in the future.

Lastly, in the future, conducting clinical trials for the prototype evaluation in a real-world setting can help validate its accuracy and impact on clinical decision-making. Usability testing with medical professionals can provide valuable feedback for the prototype performance.

5.5 Conclusion

In conclusion, this project successfully achieved all the required project objectives. The first objective of this project is to investigate the implementation of the decision tree and support vector machine (SVM) algorithms for medical diagnosis and treatment plan recommendation for myocardial infarction patients. The project identified the knowledge acquisition regarding the decision tree and support vector machines (SVM). The second objective of this project is to develop a prototype that will provide accurate diagnosis and treatment plans for myocardial patients based on the patient's medical history using the decision tree and support vector machines (SVM) algorithms have also been successfully acquired. The implementation of support vector machine (SVM) to classify ECG images and decision tree in classifying myocardial infarction risk level showcased the algorithms' abilities in handling complex data. Moreover, the third objective which is to evaluate the the performance of the decision tree and support vector machines (SVM) in diagnosing and recommending treatment plans for myocardial infarction patients which the project excelled whereby measuring using evaluation metrics including accuracy, precision, recall and F1-score resulting in high performance percentage showcasing the algorithms effectiveness.

REFERENCES

Cardiovascular diseases (CVDs). (11 June, 2021). Retrieved from World Health Organization: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)?gad_source=1&gclid=CjwKCAjwouexBhAuEiwAtW_Zx9C9ECHYNeqRWasfnypMuqXKwm7ep0Vm-RAR5palBuz_V0tIOWccwhoC_fcQAvD_BwE

GeeksForGeeks. (21 May, 2024). Retrieved from Waterfall Model - Software Engineering: https://www.geeksforgeeks.org/waterfall-model/

Huilgol, P. (4 May, 2024). *Precision and Recall in Machine Learning*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/

Kumar, P. (3 November, 2024). *Linkedin*. Retrieved from The Importance of Recall in Machine Learning: Why It Matters and How to Improve It: https://www.linkedin.com/pulse/importance-recall-machine-learning-why-matters-how-improve-kumar-be3sc/

Ranya N. Sweis, A. J. (February, 2024). *MSD MANUAL Professional Version*. Retrieved from Acute Myocardial Infarction (MI): https://www.msdmanuals.com/professional/cardiovascular-disorders/coronary-artery-disease/acute-myocardial-infarction-mi

Abedalla, A., Fadel, A., Tuffaha, I., Al-Omari, H., Omari, M., Abdullah, M., & Al-Ayyoub, M. (2019). MTRecS-DLT: Multi-Modal Transport Recommender System using Deep Learning and Tree Models. *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 274–278. https://doi.org/10.1109/SNAMS.2019.8931864

Alowais, S. A., Alghamdi, S. S., Alsuhebany, N., Alqahtani, T., Alshaya, A. I., Almohareb, S. N., Aldairem, A., Alrashed, M., Bin Saleh, K., Badreldin, H. A., Al Yami, M. S., Al Harbi, S., & Albekairy, A. M. (2023). Revolutionizing healthcare: the role of artificial intelligence in clinical practice. *BMC Medical Education*, *23*(1), 689. https://doi.org/10.1186/s12909-023-04698-z

Aryuni, M., Miranda, E., Bernando, C., & Hartanto, A. (2021). Coronary Artery Disease Prediction Model using CART and SVM: A Comparative Study. *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, *1*, 72–75. https://doi.org/10.1109/ICCSAI53272.2021.9609721

Bemando, C., Miranda, E., & Aryuni, M. (2021). Machine-Learning-Based Prediction Models of Coronary Heart Disease Using Naïve Bayes and Random Forest Algorithms. *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, 232–237. https://doi.org/10.1109/ICSECS52883.2021.00049

Bera, P. (2024). Deep learning approach for MI detection using SVM-based classifier. *2024 3rd International Conference on Power Electronics and IoT Applications in Renewable Energy and Its Control (PARC)*, 449–452. https://doi.org/10.1109/PARC59193.2024.10486283

Beyrami Enanlou, H., & Lotfivand, N. (2017). Shannon's Energy Based Algorithm in ECG Signal Processing. *Computational and Mathematical Methods in Medicine*, *2017*, 1–16. https://doi.org/10.1155/2017/8081361

Bharathi, N. N., Shaik, M. F., Poojita, T., Sravanthi, T., Rafi, M., & Raja, I. R. (2023). Heart Attack Prediction Using Artificial Neural Networks. *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, *1*, 304–308. https://doi.org/10.1109/ICACCS57279.2023.10112664

Charan, K. S., Chokkalingam, S., & Sundari, K. K. S. (2022). Efficiency of Decision tree algorithm for Brain Tumor MRI Images comparing with SVM Algorithm. *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, 1–4. https://doi.org/10.1109/MACS56771.2022.10022493

Gozali, A. A. (2023). Hypertension Multi-Year Prediction and Risk Factors Analysis Using Decision Tree. *2023 10th International Conference on Information Technology, Computer, and Electrical Engineering*

(ICITACEE), 76–82. https://doi.org/10.1109/ICITACEE58587.2023.10277644

Guabassi, I. El, Achhab, M. A., Jellouli, I., & Mohajir, B. E. El. (2016). Recommender system for ubiquitous learning based on decision tree. *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, 535–540. https://doi.org/10.1109/CIST.2016.7805107

Hafeez, M., Rashid, M., Tariq, H., Abideen, Z., Alotaibi, S., & Sinky, M. (2021). Performance Improvement of Decision Tree: A Robust Classifier Using Tabu Search Algorithm. *Applied Sciences*, *11*, 6728. https://doi.org/10.3390/app11156728

Hambali, M., Yakub, S., Oladele, T., & Gbolagade, M. (2019). *ADABOOST ENSEMBLE ALGORITHMS FOR BREAST CANCER CLASSIFICATION*. 1–10.

Harimoorthy, K., & T, M. (2021). Multi-disease prediction model using improved SVM-radial bias technique in healthcare monitoring system. *Journal of Ambient Intelligence and Humanized Computing*, *12*. https://doi.org/10.1007/s12652-019-01652-0

Ibrahim, S., Salhab, N., & Falou, A. E. (2023). Heart disease Prediction using Machine Learning. *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*, 1–6. https://doi.org/10.1109/ICAISC56366.2023.10085522

Kimura, S., Tokuhisa, M., Uemura, A., & Okada, M. (2019). Feature Selection based on Random Forests and its Application to a Modeling of Sewage Treatment Plant. *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 1–6. https://doi.org/10.1109/TAAI48200.2019.8959859

Kolikipogu, R., Darak, V., Yennapu, R., Reddy, S., Sureddi, R. M. K., & Kuchipudi, R. (2023). Agriculture Recommender System for Precision Farming using Machine Learning(ARS). *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 921–927. https://doi.org/10.1109/ICIMIA60377.2023.10426510

Kumar, A., Mehla, V. K., Tomar, H., Kumar, M., & Komaragiri, R. (2020). Classification of Normal and Abnormal ECG signals using Support Vector Machine and Fourier Decomposition Method. *2020 IEEE International Symposium on Smart Electronic Systems (ISES) (Formerly INiS)*, 161–166. https://doi.org/10.1109/iSES50453.2020.00044

Kumar, A., Rathor, K., Vaddi, S., Patel, D., Vanjarapu, P., & Maddi, M. (2022). ECG Based Early Heart Attack Prediction Using Neural Networks. *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 1080–1083. https://doi.org/10.1109/ICESC54411.2022.9885448

Kutrani, H., & Eltalhi, S. (2022). Decision Tree Algorithms for Predictive Modeling in Breast Cancer Treatment. *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 223–227. https://doi.org/10.1109/MI-STA54861.2022.9837762

Laseno, F. U. D., & Hendradjaya, B. (2019). Knowledge-Based Filtering Recommender System to Propose Design Elements of Serious Game. *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, 158–163. https://doi.org/10.1109/ICEEI47359.2019.8988797

Liu, J., Chen, Y., & Li, J. (2023). Prediction Model of Heart Attack Based on Optimized Decision Tree Algorithm. *2023 2nd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI)*, 1–6. https://doi.org/10.1109/ICDACAI59742.2023.00006

Lutimath, N. M., Sharma, N., & Byregowda, B. K. (2021). Prediction of Heart Disease using Random Forest. *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*, 1–4. https://doi.org/10.1109/ETI4.051663.2021.9619208

Mahin, M. A., & Ahmad, M. (2021). Diagnosis of Left Ventricular Hypertrophy from ECG Signals Based on CCS Methodology using SVM. *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 1–6. https://doi.org/10.1109/ICEEICT53905.2021.9667877

Manoj, M. S., Madhuri, K., Anusha, K., & Sree, K. U. (2023). Design and Analysis of Heart Attack Prediction System Using ML. *2023 IEEE*

International Conference on Integrated Circuits and Communication Systems (ICICACS), 1–6. https://doi.org/10.1109/ICICACS57338.2023.10099819

Mantle, A. (2021, July 12). *The medcolegal challenges of myocardial infarction*. Medicolegal Partners Limited.

Martínez-Gorospe, A., Sánchez-Lucero, E., Pedroza-Méndez, B. E., Quintero-Flores, P. M., Hernández-Hernández, J. C., & Sánchez-Navarro, J. C. (2021). Food recommender system based on weighted ingredients, body mass index and allergies; using the Random Forest algorithm. *2021 Mexican International Conference on Computer Science (ENC)*, 1–7. https://doi.org/10.1109/ENC53357.2021.9534806

Massoudi, M., Ghory, S., & Massoudi, M. (2021). Career Recommender System Using Decision Trees. *2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 1–4. https://doi.org/10.1109/SMARTGENCON51891.2021.9645805

Mohebbi, N., Alavi, M., Kargari, M., Mirbehbahani, S. H., & Kharazmy, A. B. (2023). A drug recommender system Based on Collaborative Filtering for Covid-19 patients. *2023 28th International Computer Conference, Computer Society of Iran (CSICC)*, 1–5. https://doi.org/10.1109/CSICC58665.2023.10105347

Pereira, S., & Karia, D. (2018). Prediction of Sudden Cardiac Death using Classification and Regression Tree Model with Coalesced based ECG and Clinical Data. *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, 678–681. https://doi.org/10.1109/CESYS.2018.8723979

Prakash, Y. N. V. S., Sathyam, B., Venkat, M. S., Rao, B. P. V. M., Naik, M. M., & Panchikkil, S. (2024). Heart Attack Detection using Machine Learning. *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, *2*, 1–6. https://doi.org/10.1109/IATMSI60426.2024.10502469

Priya, K. D., Samyogitha, A. S., Reddy, A. V. K., & Sri, B. D. (2023). ENSEMBLED CROPIFY – Crop & Fertilizer Recommender System with Leaf Disease Prediction. *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, 600–604. https://doi.org/10.1109/ICIDCA56705.2023.10100117

Rahman, S., Hasan, M. M., & Sarkar, A. K. (2022). Machine Learning and Deep Neural Network Techniques for Heart Disease Prediction. *2022 25th International Conference on Computer and Information Technology (ICCIT)*, 1086–1091. https://doi.org/10.1109/ICCIT57492.2022.10055902

Rajendran, V. G., Jayalalitha, S., Thalaimalaichamy, M., & Raj, T. N. (2021). Classification of heart disease from ECG signals using Machine Learning. *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, 606–609. https://doi.org/10.1109/RTEICT52294.2021.9573659

Rambe, M. F., & Mandala, S. (2023). Myocardial Infarction Detection as an Element of Preventive Medicine with Random Forest. *2023 International Conference on Advancement in Data Science, E-Learning and Information System (ICADEIS)*, 1–5. https://doi.org/10.1109/ICADEIS58666.2023.10270922

Reddy, K., Khaliq, A., & Henning, R. J. (2015). Recent advances in the diagnosis and treatment of acute myocardial infarction. *World J Cardiol*, *7*(5), 243–276. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4438466/

Rikendra, Kumara, S., & Kumar, K. (2023). Comparative Study on Heart Disease Prediction Using Machine Learning. *2023 International Conference on Disruptive Technologies (ICDT)*, 478–481. https://doi.org/10.1109/ICDT57929.2023.10150734

Salari, N., Morddarvanjoghi, F., Abdolmaleki, A., Rasoulpoor, S., Khaleghi, A. A., Hezarkhani, L. A., Shohaimi, S., & Mohammadi, M. (2023). The global prevalence of myocardial infarction: a systematic review and meta-analysis. *BMC Cardiovascular Disorders*, *23*(1), 206. https://doi.org/10.1186/s12872-023-03231-w

Sang, X., Yao, Q. Z., Ma, L., Cai, H. W., & Luo, P. (2020). Study on survival prediction of patients with heart failure based on support vector machine algorithm. *2020 International Conference on Robots & Intelligent System (ICRIS)*, 636–639. https://doi.org/10.1109/ICRIS52159.2020.00160

Saxena, A., Kumar, M., Tyagi, P., Sikarwar, K., & Pathak, A. (2022). Machine Learning based selection of Myocardial Complications to Predict Heart Attack. *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 1–4. https://doi.org/10.1109/UPCON56432.2022.9986434

Shanbhag, A. A., Shetty, C., Ananth, A., Shetty, A. S., Nayak, K. K., & Rakshitha, B. R. (2021). Heart Attack Probability Analysis Using Machine Learning. *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 301–306. https://doi.org/10.1109/DISCOVER52564.2021.9663631

Sharma, O. (2023). Prediction and Analysis of Heart Attack using Various Machine Learning Algorithms. *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, 786–790. https://doi.org/10.1109/AISC56616.2023.10085460

Silpa, C., Sravani, B., Vinay, D., Mounika, C., & Poorvitha, K. (2023). Drug Recommendation System in Medical Emergencies using Machine Learning. *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, 107–112. https://doi.org/10.1109/ICIDCA56705.2023.10099607

Singh, V., Chaudhuri, S., & K, A. (2023). An Advancement in a Prediction Model for Cervical Cancer Using the Cart Algorithm. *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, 1–5. https://doi.org/10.1109/ICTBIG59752.2023.10456163

Taggart, C., Wereski, R., Mills, N. L., & Chapman, A. R. (2021). Diagnosis, Investigation and Management of Patients with Acute and Chronic Myocardial Injury. *Journal of Clinical Medicine*, *10*(11). https://doi.org/10.3390/jcm10112331

Turnip, R., Nurjanah, D., & Kusumo, D. S. (2017). Hybrid recommender system for learning material using content-based filtering and collaborative filtering with good learners' rating. *2017 IEEE Conference on E-Learning, e-Management and e-Services (IC3e)*, 61–66. https://doi.org/10.1109/IC3e.2017.8409239

Wang, X. (2022). College Student Employment Management Recommendation System Based on Decision Tree Algorithm. *2022 International Conference on Education, Network and Information Technology (ICENIT)*, 169–173. https://doi.org/10.1109/ICENIT57306.2022.00044

Yanik, H., & Cömert, M. (2019). Wart Treatment Selection with a Decision Tree-Based Approach. *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 1–6. https://doi.org/10.1109/IDAP.2019.8875915

Zhao, Y., Ye, H., Zhong, J., & Wang, H. (2019). A Decision Tree Based Method for Treatment Therapy of HCC. *2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*, 879–883. https://doi.org/10.1109/SAFEPROCESS45799.2019.9213252

APPENDICES

Table 10 Input attributes in myocardial infarction complications dataset

No.	Attributes	Values
1.	Age (AGE)	Minimum: 26 Maximum: 92
2.	Gender (SEX)	0: Female 1: Male
3.	Quantity of myocardial infarctions in the anamnesis (INF_ANAM)	0: Never 1: 1 time 2: 2 times 3+ : 3 or above
4.	Exertional angina pectoris in the anamnesis	0 1 2 3 4 5 6
5.	Functional class (FC) of angina pectoris in the last year (FK_STENOK)	0: there is no angina pectoris 1: I FC 2: II FC 3: III FC 4: IV FC
6.	Coronary heart disease (CHD) in recent weeks, days before admission to hospital (IBS_POST)	0: No CHD 1: Exertional angina pectoris 2: Unstable angina pectoris
7.	Heredity on CHD (IBS_NASL)	0: Not burdened 1: Burdened
8.	Presence of essential hypertension (GB)	0: No essential hypertension 1: Stage 1 2: Stage 2 3: Stage 3
9.	Symptomatic hypertension (SIM_GIPERT)	0: No 1: Yes
10.	Duration of arterial hypertension (DLIT_AG)	0: No arterial hypertension 1: One year 2: 2 years 3: 3 years

		4: 4 years 5: 5 years 6: 6-10 years
11.	Presence of chronic heart failure in the anamnesis (ZSN_A)	0: No chronic heart failure 1: I stage 2: IIA stage (heart failure due to right ventricular systolic dysfunction) 3: IIA stage (heart failure due to left ventricular systolic dysfunction. 4: IIB stage (heart failure due to left and right ventricular systolic dysfunction)
12.	Observing of arrhythmia in the anamnesis (nr11)	0: No 1: Yes
13.	Premature atrial contractions in the anamnesis (nr01)	0: No 1: Yes
14.	Premature ventricular contractions in the anamnesis (nr02)	0: No 1: Yes
15.	Paroxysms of atrial fibrillation in the anamnesis (nr03)	0: No 1: Yes
16.	A persistent form of atrial fibrillation in the anamnesis (nr04)	0: No 1: Yes
17.	Ventricular fibrillation in the anamnesis (nr07)	0: No 1: Yes
18.	Ventricular paroxysmal tachycardia in the anamnesis	0: No 1: Yes
19.	First-degree AV block in the anamnesis (np01)	0: No 1: Yes
20.	Third-degree AV block in the anamnesis (np04)	0: No 1: Yes
21.	LBBB (anterior branch) in the anamnesis (np05)	0: No 1: Yes
22.	Incomplete LBBB in the anamnesis (np07)	0: No 1: Yes
23.	Complete RBBB in the anamnesis	0: No 1: Yes
24.	Diabetes mellitus in the anamnesis (endocr_01)	0: No

		1: Yes
25.	Obesity in the anamnesis (endocr_02)	0: No 1: Yes
26.	Thyrotoxicosis in the anamnesis (endocr_03)	0: No 1: Yes
27.	Chronic bronchitis in the anamnesis (zab_leg_01)	0: No 1: Yes
28.	Obstructive chronic bronchitis in the anamnesis (zab_leg_02)	0: No 1: Yes
30.	Bronchial asthma in the anamnesis (zab_leg_03)	0: No 1: Yes
31.	Pulmonary tuberculosis in the anamnesis (zab_leg_06)	0: No 1: Yes
32.	Systolic blood pressure according to Emergency Cardiology Team (S_AD_KBRIG)	Minimum: 0 Maximum: 260
33.	Diastolic blood pressure according to Emergency Cardiology Team (D_AD_KBRIG)	Minimum: 0 Maximum: 190
34.	Systolic blood pressure according to intensive care unit (S_AD_ORIT)	Minimum: 0 Maximum: 260
35.	Diastolic blood pressure according to intensive care unit (D_AD_ORIT)	Minimum: 0 Maximum: 190
36.	Pulmonary edema at the time of admission to intensive care unit (O_L_POST)	0: No 1: Yes
37.	Cardiogenic shock at the time of admission to intensive care unit (K_SH_POST)	0: No 1: Yes
38.	Paroxysms of atrial fibrillation at the time of admission to intensive care unit (MP_TP_POST)	0: No 1: Yes
39.	Paroxysms of supraventricular tachycardia at the time of admission to intensive care unit, (or at a pre-hospital stage) (SVT_POST)	0: No 1: Yes
40.	Paroxysms of ventricular tachycardia at the time of admission to intensive care unit, (or at a pre-hospital stage) (GT_POST)	0: No 1: Yes
41.	Ventricular fibrillation at the time of admission to intensive care unit, (or at a pre-hospital stage) (FIB_G_POST)	0: No 1: Yes
42.	Presence of an anterior myocardial infarction (left ventricular) (ECG changes in leads V1 – V4) (ant_im)	0: No infarction 1: QRS has no changes 2: QRS is like QR-complex

		3: QRS is like Qr-complex 4: QRS is like QS-complex
43.	Presence of a lateral myocardial infarction (left ventricular) (ECG changes in leads V5 – V6, I, AVL) (lat_im)	0: No infarction 1: QRS has no changes 2: QRS is like QR-complex 3: QRS is like Qr-complex 4: QRS is like QS-complex
44.	Presence of an inferior myocardial infarction (left ventricular) (ECG changes in leads III, AVF, II). (inf_im)	0: No infarction 1: QRS has no changes 2: QRS is like QR-complex 3: QRS is like Qr-complex 4: QRS is like QS-complex
45.	Presence of a posterior myocardial infarction (left ventricular) (ECG changes in V7 – V9, reciprocity changes in leads V1 – V3) (post_im)	0: No infarction 1: QRS has no changes 2: QRS is like QR-complex 3: QRS is like Qr-complex 4: QRS is like QS-complex
46.	Presence of a right ventricular myocardial infarction (IM_PG_P)	0: No 1: Yes
47.	ECG rhythm at the time of admission to hospital – sinus (with a heart rate 60-90) (ritm_ecg_p_01)	0: No 1: Yes
48.	ECG rhythm at the time of admission to hospital – atrial fibrillation (ritm_ecg_p_02)	0: No 1: Yes
49.	ECG rhythm at the time of admission to hospital – atrial (ritm_ecg_p_04)	0: No 1: Yes
50.	ECG rhythm at the time of admission to hospital – idioventricular (ritm_ecg_p_06)	0: No 1: Yes
51.	ECG rhythm at the time of admission to hospital – sinus with a heart rate above 90 (tachycardia) (ritm_ecg_p_07)	0: No 1: Yes
52.	ECG rhythm at the time of admission to hospital – sinus with a heart rate below 60 (bradycardia) (ritm_ecg_p_08)	0: No 1: Yes
53.	Premature atrial contractions on ECG at the time of admission to hospital (n_r_ecg_p_01)	0: No 1: Yes
54.	Frequent premature atrial contractions on ECG at the time of admission to hospital (n_r_ecg_p_02)	0: No 1: Yes
55.	Premature ventricular contractions on ECG at the time of admission to hospital (n_r_ecg_p_03)	0: No 1: Yes

56.	Frequent premature ventricular contractions on ECG at the time of admission to hospital (n_r_ecg_p_04)	0: No 1: Yes
57.	Paroxysms of atrial fibrillation on ECG at the time of admission to hospital (n_r_ecg_p_05)	0: No 1: Yes
58.	Persistent form of atrial fibrillation on ECG at the time of admission to hospital (n_r_ecg_p_06)	0: No 1: Yes
59.	Paroxysms of supraventricular tachycardia on ECG at the time of admission to hospital (n_r_ecg_p_08)	0: No 1: Yes
60.	Paroxysms of ventricular tachycardia on ECG at the time of admission to hospital (n_r_ecg_p_09)	0: No 1: Yes
61.	Ventricular fibrillation on ECG at the time of admission to hospital (n_r_ecg_p_10)	0: No 1: Yes
62.	Sinoatrial block on ECG at the time of admission to hospital (n_p_ecg_p_01)	0: No 1: Yes
63.	First-degree AV block on ECG at the time of admission to hospital (n_p_ecg_p_03)	0: No 1: Yes
64.	Type 1 Second-degree AV block (Mobitz I/Wenckebach) on ECG at the time of admission to hospital (n_p_ecg_p_04)	0: No 1: Yes
65.	Type 2 Second-degree AV block (Mobitz II/Hay) on ECG at the time of admission to hospital (n_p_ecg_p_05)	0: No 1: Yes
66.	Third-degree AV block on ECG at the time of admission to hospital (n_p_ecg_p_06)	0: No 1: Yes
67.	LBBB (anterior branch) on ECG at the time of admission to hospital (n_p_ecg_p_07)	0: No 1: Yes
68.	LBBB (posterior branch) on ECG at the time of admission to hospital (n_p_ecg_p_08)	0: No 1: Yes
69.	Incomplete LBBB on ECG at the time of admission to hospital (n_p_ecg_p_09)	0: No 1: Yes
70.	Complete LBBB on ECG at the time of admission to hospital (n_p_ecg_p_10)	0: No 1: Yes
71.	Incomplete RBBB on ECG at the time of admission to hospital (n_p_ecg_p_11)	0: No 1: Yes
72.	Complete RBBB on ECG at the time of admission to hospital (n_p_ecg_p_12)	0: No 1: Yes
73.	Fibrinolytic therapy by Celiasum 750k IU (fibr_ter_01)	0: No 1: Yes
74.	Fibrinolytic therapy by Celiasum 1m IU (fibr_ter_02)	0: No

		1: Yes
75.	Fibrinolytic therapy by Celiasum 3m IU (fibr_ter_03)	0: No 1: Yes
76.	Fibrinolytic therapy by Streptase (fibr_ter_05)	0: No 1: Yes
77.	Fibrinolytic therapy by Celiasum 500k IU (fibr_ter_06)	0: No 1: Yes
78.	Fibrinolytic therapy by Celiasum 250k IU (fibr_ter_07)	0: No 1: Yes
79.	Fibrinolytic therapy by Streptodecase 1.5m IU (fibr_ter_08)	0: No 1: Yes
80.	Hypokalemia (< 4 mmol/L) (GIPO_K)	0: No 1: Yes
81.	Serum potassium content (K_BLOOD) (mmol/L)	Minimum: 2.3 Maximum: 8.2
82.	Increase of sodium in serum (more than 150 mmol/L) (GIPER_Na)	0: No 1: Yes
83.	Serum sodium content (Na_BLOOD) (mmol/L)	Minimum: 117 Maximum: 169
84.	Serum AlAT content (ALT_BLOOD) (IU/L)	Minimum: 0.03 Maximum: 3
85.	Serum AsAT content (AST_BLOOD) (IU/L)	Minimum: 0.04 Maximum: 2.15
86.	Serum CPK content (KFK_BLOOD) (IU/L)	Minimum: 1.2 Maximum: 3.6
87.	White blood cell count (billions per liter) (L_BLOOD)	Minimum: 2 Maximum: 27.9
88.	ESR (Erythrocyte sedimentation rate) (ROE) (мм)	Minimum: 1 Maximum: 140
89.	Time elapsed from the beginning of the attack of CHD to the hospital (TIME_B_S)	1: Less than 2 hours 2: 2-4 hours 3: 4-6 hours 4: 6-8 hours 5: 8-12 hours 6: 12-24 hours 7: More than 1 day 8: More than 2 days 9: More than 3 days

90.	Use of opioid drugs by the Emergency Cardiology Team (NA_KB)	0: No 1: Yes
91.	Use of NSAIDs by the Emergency Cardiology Team (NOT_NA_KB)	0: No 1: Yes
92.	Use of lidocaine by the Emergency Cardiology Team (LID_KB)	0: No 1: Yes
93.	Use of liquid nitrates in the ICU (NITR_S)	0: No 1: Yes
94.	Use of lidocaine in the ICU (LID_S_n)	0: No 1: Yes
95.	Use of beta-blockers in the ICU (B_BLOK_S_n)	0: No 1: Yes
96.	Use of calcium channel blockers in the ICU (ANT_CA_S_n)	0: No 1: Yes
97.	Use of a anticoagulants (heparin) in the ICU (GEPAR_S_n)	0: No 1: Yes
98.	Use of acetylsalicylic acid in the ICU (ASP_S_n)	0: No 1: Yes
99.	Use of Ticlid in the ICU (TIKL_S_n)	0: No 1: Yes
100.	Use of Trental in the ICU (TRENT_S_n)	0: No 1: Yes
101.	Relapse of the pain in the first hours of the hospital period (R_AB_1_n)	0: No relapse 1: Only one time 2: 2 times 3: 3 or more times
102.	Use of opioid drugs in the ICU in the first hours of the hospital period (NA_R_1_n)	0: No 1: Once 2: Twice 3: Three times 4: Four times
103.	Use of NSAIDs in the ICU in the first hours of the hospital period (NOT_NA_1_n)	0: No 1: Once 2: Twice 3: Three times 4: Four or more times
105.	Relapse of the pain in the second day of the hospital period (R_AB_2_n)	0: No relapse 1: Only one 2: Twice

106.	Use of opioid drugs in the ICU in the second day of the hospital period (NA_R_2_n)	0: No 1: Once 2: Twice 3: Three times
107.	Use of NSAIDs in the ICU in the second day of the hospital period (NOT_NA_2_n)	0: No 1: Once 2: Twice 3: Three times
108.	Relapse of the pain in the third day of the hospital period (R_AB_3_n)	0: No relapse 1: Only one 2: 2 times 3: 3 or more times
109.	Use of opioid drugs in the ICU in the third day of the hospital period (NA_R_3_n)	0: No 1: Once 2: Twice
110.	Use of NSAIDs in the ICU in the third day of the hospital period (NOT_NA_3_n)	0: No 1: Once 2: Twice