

Cardiovascular Disease Identification using Advanced Machine Learning Algorithm

Dr. P. F. Khaleelur Rahiman

Associate Professor, Department of Electronics and Communication Engineering, Hindusthan College of Engineering and Technology, Coimbatore

Eswar. V

Master of Communication Systems , Hindusthan College of Engineering and Technology, Coimbatore

ABSTRACT - Multimodal medical imaging has become an important approach for improving the diagnosis and understanding of various diseases. By integrating different imaging techniques, clinicians can obtain complementary information that supports more accurate clinical interpretation. These systems enable the simultaneous evaluation of functional, molecular, and structural characteristics, thereby improving the spatial and temporal consistency of diagnostic data. One of the earliest and most widely used hybrid imaging techniques in clinical practice is the combination of positron emission tomography (PET) with computed tomography (CT). In recent years, hybrid systems combining PET with magnetic resonance imaging (MRI) have also gained significant attention in both research and clinical environments.

Techniques such as PET-CT and PET-MRI provide detailed anatomical information from CT or MRI together with functional information obtained through PET tracers, including measurements of metabolic activity such as glucose uptake. In addition, certain cardiovascular conditions, including abnormal myocardial motion, valvular heart disorders, and arterial stenosis, involve rapid biological processes that require imaging methods with high temporal resolution. Modalities such as ultrasound are commonly used in these situations because of their ability to capture fast physiological changes in real time. The integration of multiple imaging modalities therefore plays a crucial role in enhancing diagnostic accuracy and improving clinical decision-making.

Keywords : Heart Disease Detection, Medical Image Processing, Machine Learning, Support Vector Machine, Cardiac Image Analysis, Feature Extraction, Image Preprocessing, Noise Removal, Data Fusion, Cardiovascular Disease Prediction, Computer-Aided Diagnosis.

CHAPTER 1 - INTRODUCTION

Cardiovascular diseases remain one of the leading causes of mortality worldwide and are often associated with several risk factors, including diabetes, hypertension, elevated cholesterol levels, and irregular heart rhythms. Early identification of these conditions is essential in order to prevent severe complications and reduce the risk of premature death. In recent years, computational methods such as data mining and machine learning have been increasingly applied to assist in the prediction and diagnosis of heart-related disorders.

Various analytical techniques have been developed to evaluate the severity and risk of cardiac disease. Algorithms such as K-Nearest Neighbors (KNN), Decision Trees (DT), Genetic Algorithms (GA), and Naïve Bayes (NB) have been widely used for classification and

prediction tasks in medical datasets. Because cardiovascular conditions involve complex interactions among multiple physiological factors, advanced computational methods are required to analyze large volumes of clinical data and extract meaningful insights.

Machine learning, a major area within artificial intelligence, enables computer systems to learn patterns from historical data and make predictions without explicit programming. These techniques are particularly valuable in healthcare applications where large datasets containing patient information must be analyzed to support clinical decision-making. Unlike traditional statistical methods, machine learning models can capture complex and nonlinear relationships among variables, thereby improving predictive performance.

Healthcare institutions generate vast amounts of patient data through electronic medical records, diagnostic tests, and monitoring devices. Analyzing this data using machine learning techniques can assist clinicians in identifying potential health risks and making more informed treatment decisions. Classification algorithms, in particular, are commonly used to categorize patients according to their risk level for developing heart disease.

Several studies have explored different machine learning approaches for the prediction and classification of cardiovascular disorders. Artificial Neural Networks (ANNs), for example, have demonstrated strong performance in modeling complex medical datasets. Multilayer Perceptron (MLP) networks trained using backpropagation have been used to predict the presence of heart disease with improved accuracy. Additionally, publicly available datasets such as the UCI heart disease dataset have been widely used to evaluate algorithms including neural networks, decision trees, support vector machines (SVM), and Naïve Bayes classifiers. Comparative studies have shown that hybrid models combining multiple techniques can achieve strong predictive performance.

Recent advancements in deep learning have further expanded the possibilities for automated cardiovascular diagnosis. Convolutional Neural Networks (CNNs), for instance, can automatically extract meaningful features from biomedical signals such as electrocardiograms (ECG). These models can analyze cardiac cycles and identify patterns associated with abnormal heart conditions.

Historically, a large portion of medical data generated by healthcare systems was not fully utilized for analytical purposes. The application of modern machine learning and deep learning techniques has significantly improved the ability to analyze this data effectively. These approaches not only enhance diagnostic accuracy but also reduce the cost and complexity of traditional diagnostic procedures. Consequently, machine learning-based

models have emerged as promising tools for improving the prediction and classification of heart diseases in modern healthcare systems.

CHAPTER 2 - LITERATURE SURVEY

2.1 Literature Survey

1. Study by Rong Tao et al. (2018)

Rong Tao and colleagues presented a machine learning-based framework for the detection and localization of ischemic heart disease using magnetocardiography (MCG) signals. In their work, the T-wave portion of averaged MCG recordings was segmented and a total of 164 features were extracted. These features were grouped into three categories: time-domain features, frequency-domain features, and information-theoretic features. Multiple classifiers such as K-Nearest Neighbor, Decision Tree, Support Vector Machine, and XGBoost were evaluated to determine the most suitable model for disease detection.

The researchers combined high-performing classifiers through an ensemble approach to improve prediction reliability. For identifying the location of ischemia, the dataset was categorized according to stenosis regions including the left anterior descending artery, left circumflex artery, and right coronary artery. An XGBoost classifier using selected time-domain features was applied for this localization task. The results demonstrated that the hybrid SVM-XGBoost model achieved strong performance with high accuracy and precision metrics. The study concluded that synchronization characteristics of T-wave repolarization and magnetic field distribution patterns are useful indicators for identifying ischemic conditions and locating affected regions in the heart.

2. Study by Yiwen Meng et al. (2019)

Yiwen Meng and co-authors investigated the use of wearable sensor data for evaluating the health status of patients with heart disease. Their research focused on individuals diagnosed with stable ischemic heart disease and utilized data collected from activity trackers. The objective was to build machine learning models capable of predicting patient-reported outcomes related to health conditions.

By analyzing physical activity patterns and sensor-generated data, the study demonstrated that machine learning algorithms could classify self-reported health status with reasonable accuracy. The findings suggest that continuous monitoring using wearable devices can support early identification of deteriorating health conditions and assist healthcare professionals in providing timely medical intervention.

3. Study by Jian Ping Li et al. (2020)

Jian Ping Li and collaborators proposed an intelligent system for heart disease detection in an e-healthcare environment using several machine learning techniques. Their model incorporated multiple classification algorithms including Support Vector Machine, Logistic Regression, Artificial Neural Networks, K-Nearest Neighbor, Naïve Bayes, and Decision Tree.

To improve model efficiency, the researchers applied different feature selection techniques such as Relief, Minimal Redundancy Maximal Relevance, Least Absolute Shrinkage and Selection Operator, and Local Learning. In addition, they introduced a novel feature selection method known as Fast Conditional Mutual Information Maximization. This approach helped eliminate redundant attributes and enhanced classification performance while reducing computational complexity.

The system was evaluated using a leave-one-subject-out cross-validation strategy to ensure reliable model assessment and hyperparameter optimization. Experimental results indicated that the proposed FCMM-SVM model achieved better prediction accuracy compared with several existing methods, demonstrating its effectiveness in identifying heart disease using clinical data.

4. Study by Haolin Wang et al. (2020)

Haolin Wang and co-researchers explored a data-driven approach for predicting resistance to intravenous immunoglobulin therapy in patients with Kawasaki disease. Their framework combined co-clustering techniques with interpretable machine learning models to address challenges such as incomplete clinical data and limited transparency of traditional algorithms.

Initially, co-clustering was applied to identify patterns in missing clinical information by simultaneously grouping patient records and clinical variables. This strategy allowed the researchers to perform group-based feature selection and develop subgroup-specific predictive models. Subsequently, group Lasso was employed to determine important risk factors within each subgroup. Finally, an Explainable Boosting Machine, which is based on generalized additive models, was utilized for prediction.

The evaluation using real-world electronic health records showed that the proposed multi-stage framework outperformed several baseline models. The study emphasized the importance of interpretable machine learning methods for healthcare applications where transparency and reliability are critical for clinical decision making.

5. Study by Pronab Ghosh et al. (2020)

Pronab Ghosh and associates presented a machine learning-based model for predicting cardiovascular disease using advanced feature selection and ensemble classification techniques. Their study utilized a combined dataset derived from multiple well-known heart disease repositories including Cleveland, Long Beach VA, Switzerland, Hungarian, and Statlog datasets.

To identify relevant attributes influencing cardiovascular risk, the authors employed feature selection approaches such as Relief and Least Absolute Shrinkage and Selection Operator. After selecting the most significant features, several ensemble classification techniques were implemented. These included Decision Tree Bagging, Random Forest Bagging, K-Nearest Neighbor Bagging, AdaBoost, and Gradient Boosting models.

The performance of these classifiers was evaluated using multiple metrics including accuracy, sensitivity, precision, F1-score, error rate, false positive rate, and false negative rate. Experimental findings indicated that ensemble models combined with effective feature selection methods can significantly enhance the prediction accuracy of cardiovascular disease detection systems.

CHAPTER 3 - PROJECT DESCRIPTION

3.1 Existing Methodology

Traditional approaches for detecting cardiac diseases often rely on devices that analyze heart sounds using ultrasonic sensors placed on the patient's chest. These sensors capture acoustic signals generated by the heart during the pumping of blood. The recorded signals are then analyzed to determine abnormalities in heart function.

However, the effectiveness of this method is limited because the diagnostic outcome mainly depends on the quality of the captured sound signals. Any disturbance or noise during signal acquisition can significantly affect the reliability of the results. In addition,

inaccuracies may arise due to environmental noise, improper sensor placement, or variations in patient conditions.

Another limitation of the existing system is the dependency on hardware components such as sensors and electronic circuits. Malfunctions or failures in these components may introduce errors into the data acquisition process, which ultimately reduces the overall accuracy and reliability of the diagnostic system. Therefore, more advanced computational techniques are required to improve diagnostic precision and reduce dependency on hardware-based measurements.

3.2 Proposed System

To overcome the limitations of the conventional approach, the proposed system utilizes digital image processing techniques combined with machine learning algorithms for improved cardiac disease detection. Instead of relying only on acoustic signals, the system analyzes cardiac diagnostic images to extract relevant features that can help identify potential heart abnormalities.

The proposed model is implemented using the MATLAB environment, where a machine learning classifier is applied to perform disease prediction. In this work, the Support Vector Machine (SVM) algorithm is employed due to its strong capability in handling classification problems and high-dimensional data.

The proposed framework includes several preprocessing steps to enhance image quality and improve classification performance. Initially, color conversion is performed to standardize the input images. After that, noise present in the images is removed using a median filtering technique, which helps preserve important structural details while eliminating unwanted distortions.

Once preprocessing is completed, feature extraction is carried out and the processed data is provided to the SVM classifier for training and prediction. Finally, a data fusion stage is applied after classification to compare neighboring pixel values and integrate information from different sources, thereby improving the reliability and accuracy of the final diagnostic output.

Overall, the proposed system aims to achieve higher prediction accuracy compared to the conventional sound-based diagnostic methods.

3.3 Project Scope

Advancements in clinical decision support systems combined with computerized patient records have significantly improved healthcare services by reducing medical errors and enhancing diagnostic precision. Integrating computational intelligence with patient data enables healthcare professionals to make more informed decisions regarding disease prediction and treatment.

The proposed system focuses on predicting the risk of heart disease by analyzing various factors associated with cardiovascular health. Historical patient data is used as the primary source for training the machine learning model. Important features such as age, gender, smoking habits, obesity, alcohol consumption, cholesterol levels, blood pressure, and heart rate are considered as input parameters for prediction.

By applying the Support Vector Machine classification algorithm to these risk factors, the system can identify patterns related to coronary heart disease. The goal of this approach is to develop a reliable prediction model that assists medical professionals in identifying high-risk patients at an early stage and improving preventive healthcare strategies.

3.4 Block Diagram

The block diagram illustrates the main stages involved in the proposed heart disease prediction system. Each stage plays an important role in processing the medical data and generating the final prediction result.

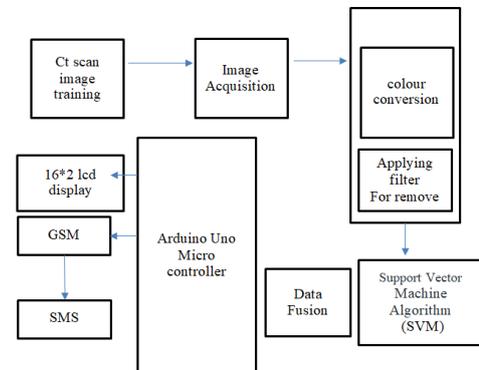


Image Acquisition

Machine learning models require large volumes of reliable and high-quality data for effective training and evaluation. Therefore, the initial stage of the system involves collecting cardiac images or relevant medical datasets. With the increasing availability of organized cardiac imaging datasets, researchers can develop robust predictive models capable of performing accurate diagnosis and classification.

Preprocessing

Preprocessing is an essential step in preparing raw data for analysis. In this stage, the collected data is cleaned and transformed into a format suitable for machine learning algorithms. Various preprocessing techniques such as handling missing values, normalization, and scaling methods like standard scaling or Min–Max scaling are applied to improve data quality. Proper preprocessing ensures that the dataset becomes more consistent and suitable for classification tasks.

Thresholding

Thresholding is an image processing technique used to enhance image contrast and improve visual clarity. This process converts grayscale images into binary representations by assigning pixel values based on predefined threshold levels. Pixel intensity values may range up to 256 in grayscale images, and selecting an appropriate threshold value helps distinguish relevant structures from background noise. This step facilitates easier analysis of important image features.

Feature Selection using Support Vector Machine (SVM)

The Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for classification and regression tasks. In classification problems, SVM attempts to find an optimal boundary known as a hyperplane that separates data points belonging to different classes.

Each data sample is represented as a point in an n-dimensional feature space, where n corresponds to the number of input features. The SVM algorithm determines the hyperplane that maximizes the margin between different classes, ensuring better generalization and classification accuracy.

Data points that lie closest to the decision boundary are known as support vectors, and they play a critical role in defining the position of the hyperplane. By maximizing the margin between classes,

SVM provides a robust classification model capable of handling complex datasets.

In cases where the data is not linearly separable, SVM uses a technique called the kernel trick. Kernel functions transform the input data into a higher-dimensional feature space, making it easier to separate the classes using a linear hyperplane. This approach allows SVM to effectively solve non-linear classification problems.

Noise Removal using Image Filtering

Digital images often contain noise introduced during image acquisition or transmission. Common types of noise include Gaussian noise, Poisson noise, speckle noise, and salt-and-pepper noise. These distortions may reduce image quality and affect the performance of image analysis algorithms.

To address this issue, filtering techniques are applied to remove noise before further processing. Several filtering methods such as minimum filters, maximum filters, and median filters can be used. Among these, the median filter is widely preferred because it effectively removes noise while preserving important edges and structural details in the image.

Data Fusion

The electrocardiogram (ECG) signal contains valuable information about the electrical activity of the heart. Characteristics such as the shape and duration of the P, QRS, and T waves, as well as the RR interval, provide important insights into cardiac health. However, relying on a single signal source may not provide sufficient information for accurate diagnosis.

Therefore, combining multiple physiological signals such as ECG, blood pressure, oxygen saturation levels, and respiratory signals can significantly improve diagnostic accuracy. Data fusion techniques integrate these heterogeneous data sources and analyze them collectively to detect life-threatening cardiac conditions.

In this project, a computational approach based on fuzzy logic is used to combine information from multiple signals. This approach helps in evaluating the severity of heart disease by calculating a parameter known as the deterioration index, which provides a quantitative measure of patient health status.

CHAPTER 4 - SYSTEM IMPLEMENTATION AND TESTING

4.1 Hardware Requirements

The hardware components used in the proposed system are essential for executing the image processing and machine learning algorithms effectively. The system is designed to operate on a standard computing platform capable of handling medical image datasets and performing numerical computations.

Typical hardware requirements include:

- **Processor:** Intel Core i3 / i5 or higher
- **RAM:** Minimum 4 GB (8 GB recommended for large datasets)
- **Storage:** At least 500 GB hard disk
- **Display:** Standard monitor for visualization of processed images
- **Input Devices:** Keyboard and mouse for system interaction

These hardware resources ensure that the system can efficiently execute MATLAB-based image processing and machine learning operations.

4.1 System Testing

System testing is an important stage in the software development process. It consists of a set of structured activities designed to evaluate whether the developed system functions according to the specified requirements. Testing must be carefully planned and executed to detect errors and ensure that the system operates correctly under different conditions.

Since testing often requires significant time and effort during software development, it is necessary to follow a systematic strategy. Proper testing ensures that defects are identified early and that the final software product meets the expected quality standards.

4.1.1 Testing Objectives

The primary objectives of system testing include:

- Verifying that the software meets the specified functional requirements
- Identifying errors or defects present in the system
- Ensuring that all modules interact correctly with each other
- Evaluating system performance under various input conditions
- Confirming that the software produces accurate outputs

Testing helps improve system reliability and ensures that the developed application performs efficiently in real-world scenarios.

Types of Testing

Software testing can be broadly classified into two categories:

1. **Unconventional Testing**
2. **Conventional Testing**

Unconventional Testing

Unconventional testing is generally carried out by the Software Quality Assurance (SQA) team. This testing approach focuses on verifying the development process throughout the entire project lifecycle. The main goal is to ensure that the project development activities comply with the defined standards and client requirements.

In this process, the SQA team evaluates different aspects of the project, including design, coding practices, and documentation. The following techniques are commonly used:

- **Peer Review:** Team members evaluate each other's work to identify errors and improve code quality.
- **Code Walkthrough:** Developers explain the logic of their code to other team members for review and feedback.
- **Inspection:** A formal examination of software artifacts such as design documents and source code.
- **Document Verification:** Ensuring that project documentation meets required standards and accurately reflects system functionality.

Conventional Testing

Conventional testing focuses on detecting defects in the developed system and verifying that the software meets client requirements.

In this approach, the testing team executes the application using various input conditions and observes the outputs.

If any defects or unexpected results are found, they are reported to the development team for correction. This process helps improve system stability and ensures that the final application functions correctly.

4.1.2 Test Case Design

Unit Testing

Unit testing is the process of evaluating individual components of the software independently. Each module is tested separately to ensure that it performs its intended function correctly before being integrated with other modules.

This testing method helps detect errors at an early stage of development. During unit testing, different input scenarios are applied to verify system behavior. For example:

- Checking system response to empty input fields
- Testing invalid data formats such as incorrect email addresses or URLs
- Verifying that duplicate entries are not accepted
- Ensuring proper validation of date formats and user credentials

Unit testing significantly reduces the number of defects that appear in later stages of development.

Integration Testing

Integration testing is performed after individual modules have been successfully tested. In this stage, the modules are combined and tested as a group to verify their interactions.

The purpose of integration testing is to ensure that the modules communicate correctly and that data flows properly between them. Test cases are designed to confirm that the integrated system operates successfully under different conditions.

Validation Testing

Validation testing ensures that the developed system meets the expectations of the end users. This stage verifies that the software performs the tasks required by the users and satisfies the specified functional requirements.

In many cases, validation testing includes **Alpha Testing** and **Beta Testing**, where the system is evaluated by real users. Their feedback helps identify issues that may not have been detected during earlier testing stages.

4.1.3 Testing Strategies

Several testing strategies have been proposed to guide developers during the software testing process. Although these strategies may differ in implementation, they share several common principles:

- Testing begins with individual components and gradually progresses toward the complete system.
- Different testing techniques are applied at various stages of development.
- Developers usually perform testing initially, while larger projects may involve independent testing teams.
- Testing and debugging are separate activities, although debugging is performed after defects are detected during testing.

Integration Testing Strategy

Integration strategies determine how different software modules are combined during the development process. Common approaches include:

- **Top-Down Integration:** Higher-level modules are tested first and lower-level modules are integrated later.
- **Bottom-Up Integration:** Lower-level modules are tested first before integrating higher-level components.

These strategies help ensure that all modules work together effectively to produce the final system.

White Box Testing

White box testing involves examining the internal structure and logic of the software. In this approach, testers analyze the program code, internal variables, and execution paths to ensure correct implementation of algorithms.

This testing method provides a clear understanding of system behavior during execution and helps identify logical errors in the code.

Black Box Testing

Black box testing focuses on evaluating the functionality of the software without considering its internal structure. Testers provide input values and observe the corresponding outputs to determine whether the system behaves as expected.

This method is useful for verifying whether the software meets user requirements.

Interface Testing

Interface testing verifies communication between different modules or components of the system. This testing ensures that data is correctly exchanged between subsystems during integration.

Module Testing

Module testing involves testing each functional module of the system independently. The goal is to ensure that each module produces correct outputs for the given inputs. This approach helps identify errors before integrating modules into the final system.

Maintenance

Maintenance is the final stage of the software development lifecycle. After deployment, the system must be continuously monitored and updated to adapt to changing requirements or technological advancements.

Maintenance activities may include:

- Correcting errors that appear during real-world operation
- Improving system performance
- Updating the system to support new requirements

A well-designed system should allow modifications without affecting other components. This ensures that the system remains stable and accurate even after updates.

4.2 System Implementation

4.2.1 Software Overview

The proposed system is implemented using **MATLAB**, a high-level programming environment widely used for numerical computation, data analysis, and algorithm development.

MATLAB provides powerful tools for:

- Numerical computing
- Data visualization
- Algorithm development
- Model simulation

The platform enables researchers and engineers to analyze data, design algorithms, and create applications more efficiently compared to traditional programming languages such as C/C++ or Java.

Key Features of MATLAB

Important capabilities of MATLAB include:

- High-level programming language for numerical computation
- Interactive environment for problem solving and algorithm design
- Extensive mathematical libraries for linear algebra and statistics
- Built-in visualization tools for plotting and graphical representation
- Development tools for debugging and performance optimization
- Integration capabilities with other programming languages and applications

4.2.2 Numerical Computation

MATLAB offers a wide range of numerical methods for data analysis and scientific computing. The platform includes built-in mathematical functions that support many engineering and scientific operations.

Common numerical computation techniques include:

- Interpolation and regression analysis
- Differentiation and integration
- Solving linear equations
- Fourier analysis
- Eigenvalue and singular value calculations
- Solving ordinary differential equations
- Handling sparse matrices

Additional MATLAB toolboxes provide advanced capabilities for statistics, optimization, signal processing, and machine learning.

Data Analysis and Visualization

MATLAB includes powerful tools for analyzing and visualizing data. These tools allow users to explore datasets, identify patterns, and present results using graphical representations such as charts, plots, and diagrams.

Users can also generate reports that include program code, results, and visualizations, which can be exported in formats such as PDF, Word, or HTML.

4.2.3 The MATLAB Language

The MATLAB programming language is specifically designed for matrix and vector operations, which are essential for solving many engineering and scientific problems.

Unlike traditional programming languages, MATLAB automatically manages memory allocation and data types, enabling faster program development. Many complex calculations can be performed using a single line of MATLAB code.

The language also supports common programming constructs such as:

- Conditional statements
- Loops
- Error handling
- Object-oriented programming

MATLAB provides development tools such as the Command Window, Code Editor, and Debugging tools that help developers create efficient and maintainable programs.

Image Processing Toolbox

MATLAB includes an Image Processing Toolbox that provides advanced functions for analyzing and manipulating images. This toolbox supports several image formats, including:

- JPEG
- PNG
- TIFF
- HDF
- DICOM medical images

These capabilities make MATLAB suitable for medical image analysis applications.

Image Preprocessing and Enhancement

Image preprocessing techniques are applied to improve image quality before performing analysis. These techniques help reduce noise, improve contrast, and highlight important features.

Common image enhancement methods include:

- Histogram equalization
- Contrast stretching
- Gamma correction
- Linear and median filtering

Filtering techniques are used to remove noise while preserving important details within the image.

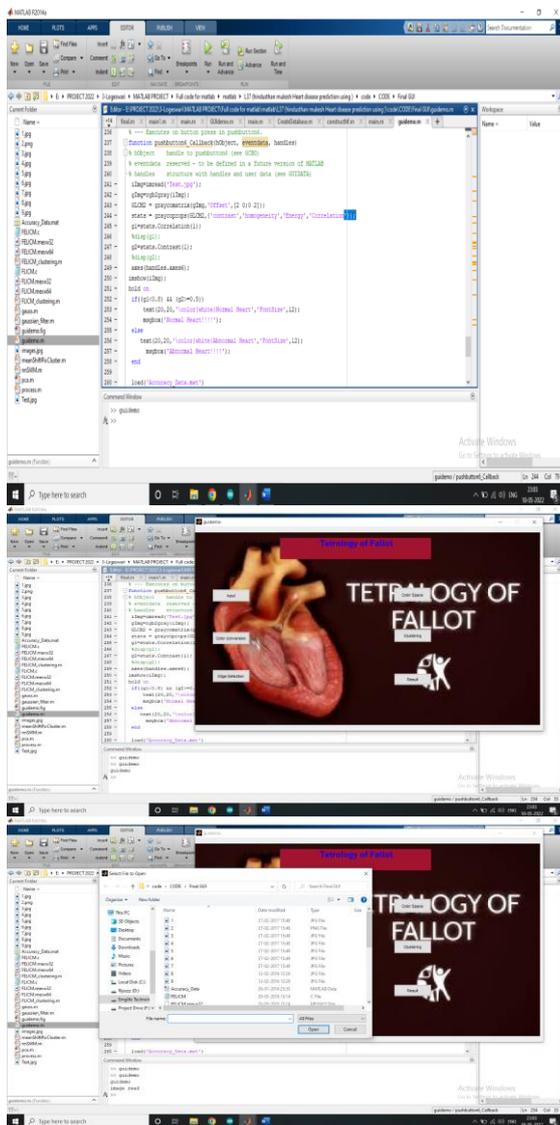
CHAPTER 5

5.1 CONCLUSION

Determining any heart disease on some raw data is really difficult for even a doctor which is why many healthcare sectors are opting for machine learning techniques to determine it. In our experimental study we took the Cleveland Heart Disease dataset obtained from UCI repository and applied pre-processing to drop the data with missing values and applied some algorithm like Decision tree, K-Nearest Neighbour, Support Vector Machines and, Random Forest of which Decision Tree gives an accuracy of 79% as shown in Fig. 3, K-Nearest Neighbour gives an accuracy of

87% as shown in Fig. 4, Support vector Machines gives an accuracy of 83% as shown in Fig. 5, and Random Forest Gives an accuracy of 84% as. The ROC curve for Decision Tree, K-Nearest Neighbor, Support vector Machines By using the technique we can assure the accuracy of the result will better than the existing method. The data fusion method is must after the step of SVM method, because that will help to compare the pixel values of neighbour.

5.2 OUTPUT IMAGES



REFERENCES

[1] Anitek Bhattacharya, Mohan Mishra, Anushikha Singh & Malay Kishore Dutta, "Machine Learning Based Portable Device for Detection of Cardiac Abnormality" International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), IEEE, pp: 1-4, 2017.

[2] Dilip Kumar Choubey, Sudhakar Tripathi, Prabhat Kumar, Vaibhav Shukla, Vinay Kumar Dhandhanian, "Classification of Diabetes by Kernel based SVM with PSO", Recent Advances in Computer Science and Communications, Bentham Science, Vol. 12, No. 1, pp. 1-14, 2019.

[3] Sneha Borkar, Prof. M. N. Annadate, "Supervised Machine Learning Algorithm for Detection of Cardiac Disorders" Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE, 2018.

[4] Raid Lafta, Ji Zhang, Xiaohui Tao, Yan Li, Xiaodong Zhu, Yonglong Luo and Fulong Chen, "Coupling a Fast Fourier Transformation with a Machine Learning Ensemble Model to Support Recommendations for Heart Disease Patients in a Telehealth Environment" IEEE, pp:10674-10685, 2017

[5] Rui Chen, Aijia Lu, Jingjing Wang, Xiaohai Ma, Lei Zhao, Wanjia Wu, Zhicheng Du, Hongwen Fei, Qiongwen Lin, Zhuliang Yuf, Hui Liu "Using machine learning to predict one-year cardiovascular events in patients with severe dilated cardiomyopathy" European Journal of Radiology, Elsevier, 2019.

[6] Gunasekarn Manogaran, R. Varatharajan, M. K. Priyan, "Hybrid Recommendation System for Heart Disease Diagnosis based on Multiple Kernel Learning with Adaptive Neuro-Fuzzy Inference System" Multimedia tools and applications 77, Springer, pp: 4379-4399, 2018.

[7] W. L. Costa, L. S. Figueiredo, and E. T. A. Alves, "Application of an Artificial Neural Network for Heart Disease Diagnosis" Brazilian Congress on Biomedical Engineering, Springer, pp. 753-758, 2019.

[8] Dilip Kumar Choubey, Sanchita Paul, Smita Sandilya, Vinay Kumar Dhandhanian, "Implementation and Analysis of Classification algorithms for Diabetes", Current Medical Imaging Reviews, Bentham Science (In Press 2018) Dilip Kumar Choubey, Manish Kumar, Vaibhav Shukla, Sudhakar Tripathi, Vinay Kumar Dhandhanian, "Comparative Analysis of Classification Methods with PCA and LDA for Diabetes", Current Diabetes Reviews, Bentham Science (In Press 2020).

[9] Subhashini Narayan, E. Sathiyamoorthy, "A novel recommender system based on FFT with machine learning for predicting and identifying heart diseases" Neural Computing and Applications 31, Springer, pp: 93-102, 2019.

[10] Prerna Sharma, Krishna Choudhary, Kshitij Gupta, Rahul Chawla, Deepak Gupta, Arun Sharma, "Artificial Plant Optimization Algorithm to detect Heart Rate and Presence of Heart Disease using Machine Learning" Artificial Intelligence in Medicine, Elsevier, pp:101752, 2019.