HEART DISEASE PREDICTION USING MACHINE LEARNING APPROACH

Dissertation submitted in partial fulfillment of the requirements for the award of the degree of

Masters of Science

in

Mathematics and Computing

Submitted by

Isha Gupta

Roll No. 302203003

Under the guidance of

Dr. Anu Bajaj and Dr. Vikas Sharma

July 2024
School of Mathematics

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

Patiala – 147004
Punjab, India

CERTIFICATE

This is to certify that the thesis entitled "**Heart Disease Prediction using Machine learning approach**", being presented in partial fulfilment of the requirements for the award of degree of Masters of Science in Mathematics and Computing and submitted to the **School of Mathematics** (SOM), Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Anu Bajaj** and **Dr. Vikas Sharma**.

The matter presented in this thesis has not been submitted for the award of any other degree from this or any other institution.

Isha Gupta

Roll No. 302203003

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Dr. Anu Bajaj

Supervisor

CSED, TIET

Patiala

Dr. Vikas Sharma

Supervisor

SOM, TIET

Patiala

Acknowledgement

This thesis marks the end of the beautiful journey to achieve my Master's degree. Throughout this journey I have been supported and guided by several people. I would like to take this opportunity to express my gratitude to all those people.

My first and sincere appreciation goes to **Dr. Anu Bajaj**, my guide, for all I have learnt from her and for her continuous help and support in all stages of this thesis. Her insights and clarity of thoughts have been presented at every moment of this work. I would like to thank her for encouraging and helping me to shape my interests and ideas. It is with immense gratitude that I acknowledge **Dr. Vikas Sharma**, my co-guide. His advices and discussions were in-valuable to me and his attitude towards research always inspired me. I really appreciate him for always being so supportive.

I express my gratitude to all the faculty members and staff of the School of Mathematics, Thapar Institute of Engineering and Technology, for their support. Above all I would like to thank my parents for their love, blessings, support, encouragement, sacrifice, and unwavering belief in me. Without them, I would not be the person I am today. I thank and pay my regards to the Almighty for his love and blessings.

Date: 30/07/2024 Isha Gupta

Place: Patiala Roll No. 302203003

Abstract

Heart diseases have become the primary cause of death globally. Therefore, it is essential to develop robust diagnostic and treatment methods. This thesis focuses on diagnosing heart disorders. We utilized the MIT-BIH Arrhythmia Dataset to conduct a comparative analysis of various machine learning (ML) techniques, including Random Forest (RF), K-Nearest Neighbor (KNN), and Decision Tree (DT), along with deep learning (DL) models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). To enhance predictive performance, various preprocessing methods were employed, including filtering, normalization, and comprehensive feature selection techniques like chi-square and sequential feature selector.

Additionally, an advanced prediction was proposed, combining feature selection using a hybrid of Genetic Algorithm (GA) and Cuckoo Search Optimization (CSO) with a majority voting ensemble of Convolutional Neural Network and Random Forest on UCI Heart disease dataset. This approach also integrated GA for hyperparameter tuning, enhancing predictive accuracy. Comprehensive preprocessing techniques were employed to ensure data quality, including handling missing values, outlier detection, and normalization. The results demonstrate that our method outperforms traditional models.

This study contributes to advancing predictive analytics in cardiovascular healthcare, aiming to support early diagnosis and informed decision-making processes through robust and accurate predictive models.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Being one of the main causes of death worldwide, heart disease poses a serious threat to healthcare systems and highlights the importance of early detection and efficient treatment methods. This thesis employs the ML algorithms for the prediction of heart disease using textual data from the UCI Heart Disease Dataset and the ECG information from the MIT-BIH Arrhythmia Dataset. It also discusses preprocessing and feature selection in data preprocessing steps.

In this chapter, we start by providing an overview of the background and motivation for this work, emphasizing the prevalence and impact of heart diseases. Next, we talk about the application of machine learning in healthcare, discussing its advantages and limitations. We then discuss the problem statement, highlighting the need for better predictive models for heart disease detection. The research objectives are outlined, detailing the specific goals of this study. Finally, we provide a roadmap of structure of thesis, summarizing the content of each chapter.

1.1 Background and Motivation

Cardiovascular diseases (CVDs) comprise a wide range of conditions affecting the heart and blood vessels. The high prevalence and mortality rate associated with heart diseases underscore the urgent need for effective diagnostic tools and preventive measures [1].

Heart diseases are influenced by a combination of physiological and behavioral risk factors [2]. Physiological factors include high blood pressure, diabetes, high cholesterol levels, and obesity. Behavioral factors encompass unhealthy eating habits, physical inactivity, smoking, and excessive alcohol consumption [4]. Traditional clinical methods for diagnosing heart disease rely heavily on static factors such as physical examinations, patient history, and conventional diagnostic tests. These methods, while valuable, often depend on the clinician's expertise and experience, leading to variability in diagnosis and treatment decisions. Moreover, traditional diagnostic procedures can be time-consuming and sometimes invasive, highlighting the need for more efficient and accurate diagnostic tools [7].

In recent years, advancements in artificial intelligence (AI), semantic computing, machine learning (ML), and deep learning (DL) have revolutionized various sectors, including healthcare. The application of these technologies in healthcare aims to enhance the accuracy and efficiency of diagnostic models, thereby improving patient outcomes. The motivation behind this research is to harness the potential of ML and DL techniques to develop robust predictive models for heart disease, ultimately aiming to lower

the mortality rate associated with cardiovascular conditions. This study focuses on utilizing textual data from the UCI Heart Disease Dataset and ECG data from the MIT-BIH Arrhythmia Dataset to create comprehensive models that can assist in heart disease prediction.

1.2 Machine Learning in Healthcare

Machine learning (ML), a branch of artificial intelligence (AI), is a fast-developing field that focuses on creating statistical models and algorithms that let computers carry out specified tasks without manual guidance. Rather, these models use past data to learn from and forecast future events. Healthcare industry is aware of the immense potential of machine learning (ML).

Advantages of Machine Learning in Healthcare

There are numerous benefits that machine learning offers to healthcare services. The ability to process huge volumes of data is one of the most noteworthy assets of the concept. Many technologies generate big data for health care organizations ranging from medical imaging, wearable technology and electronic health records. These large datasets are easily managed for processing and analysis of data using the machine learning (ML) algorithms as compared to complicated analysis done over a set of data. ML is required to identify patterns, predict the future, and help with decisions based on data.

ML can come across patterns and relations within the data that may not be easily identified by ordinary statistical methods [5]. This is very helpful for determining treatment procedures, and the diagnosis of illnesses. For instance, ML can help interpret even the slightest changes in x-rays or MRI scans that may be an indication of an early stage disease.

Moreover, through the implementation of the ML framework, one obtains non-intrusive methods of identifying diseases, thus lowering the number of invasions and increasing the comfort of patients [5]. For instance, instead of biopsies, algorithms can look at images to detect such diseases as cancer or any irregularity present in the body.

Disadvantages of Machine Learning in Healthcare

As described, integration of ML in healthcare comes with numerous advantages; nonetheless, there are also liabilities. Availability and quality of the data is one of the main issues. Availability and quality of training data govern the efficiency of the ML algorithms because they are fed on these data. The accuracy of ML models can be adversely affected by the poor, irregular or random data which is frequently encountered in the healthcare systems. However, there is a key factor that is standing in the way of exploitation of the full potential of machine learning in the health care sector and this is the issue of data standards and data quality.

Transition from Machine Learning to Deep Learning

Deep learning (DL), a subset of machine learning (ML), uses neural networks to extract intricate patterns, thereby mitigating some of the drawbacks of classical ML. Deep Learning (DL) simulates the neural architecture of the human brain, allowing computers to absorb and comprehend data similarly. Deep learning (DL) models are well-suited for image and signal processing applications because of their ability to handle complex data structures and non-linear correlations.

The necessity to manage increasingly complicated data and derive more significant insights has led to the shift in healthcare from machine learning to deep learning. Although machine learning techniques are effective in many tasks, they have low efficiency if dealing with high-dimensional data and non-linear and complex relations [32]. On the other hand, the DL models are capable to learn hierarchical data representations and therefore well-suited for these challenges.

Adopting DL in the healthcare sector has many benefits. Handling large amount of data is done quickly by DL models so as to enable the model to predict and analyze the data in real time. They can also decrease the burden of medical staff, enhance the accuracy of diagnostic measures, and, therefore, facilitate the treatment of patients. But the usage of DL models in some particular healthcare scenarios can be limited because of high computational requirement and the necessity of large amounts of labeled data for their training.

The adoption of ML and DL in healthcare offers numerous advantages. These technologies can process large volumes of data quickly, providing real-time insights and predictions. They can also improve the accuracy of diagnoses and treatment plans, reduce the burden on healthcare professionals, and ultimately enhance patient care. Table 1.1 provides a brief overview of common ML and DL models used in medical diagnosis.

Table 1. 1 Overview of ML models

Model	Model Name	Description
ML	Decision Tree (DT)	A tree-structured model used for classification and regression by splitting data into subsets.
ML	Extreme Gradient Boost (XGB)	A powerful algorithm that combines outputs of weak learners to yield a strong predictive model.
ML	Support Vector Machine (SVM)	Works by finding hyperplane that can separate classes in the feature space.
ML	Naïve Bayes (NB)	A model based on Bayes Theorem
DL	Long Short -Term Memory (LSTM)	Neural network capable of capturing long-term dependencies.

1.3 Problem Statement

Since heart disease is world's leading causes of death, accurate predictive models are crucial for early detection. Despite the advancements, vital aspects such as feature selection, outlier detection, and hyperparameter optimization are frequently overlooked in existing work. In order to fill this gap, this thesis compares current models before presenting a novel method that combines cutting-edge methods like min-max normalization, feature selection using a hybrid Genetic Algorithm and Cuckoo Search Algorithm (GA+CSA), and outlier detection using the Interquartile Range (IQR). The proposed strategy achieves a 95% accuracy on the UCI Heart Disease dataset by combining majority voting with Random Forest (RF) and Convolutional Neural Network (CNN) and enhancing it using GA-based hyperparameter tuning. This thesis uses a novel approach that combines deep learning algorithms with classic machine learning to increase the reliability of heart disease prediction models.

1.4 Research Objectives

The main purpose of this thesis is to build and compare the models for cardiac diseases prediction using textual information and ECG data. The specific objectives are:

1. To preprocess the raw ECG signals in order to make them fit for subsequent phases of processing and to feed to Machine Learning models.
2. In order to rank different machine learning algorithms on the set of ECG signals.
3. To preprocess UCI Heart Disease dataset in order to identify most important features.
4. To perform an evaluation on the proposed models based on the right measures of discriminability.

1.5 Structure of Thesis

This thesis is structured as follows:

- Chapter 2: Literature Review - This chapter reviews existing work on machine learning methods used in heart disease prediction.
- Chapter 3: Preliminaries – This chapter discusses all the preprocessing steps required to transform raw textual and signal data into a form suitable for further analysis. It also provides a detailed explanation of the nature inspired algorithms and machine learning models used in this thesis.
- Chapter 4: Comparative Analysis of existing methods – This chapter discuss the approach of comparing the proposed strategy with existing Machine learning, deep learning and hybrid algorithms applied on the MIT-BIH signal data followed by the results and finding.

- Chapter 5: Proposed Methodology - This chapter presents the specific flow of the study involving pre-processing, feature selection, model development and model parameter selection on the UCI heart disease dataset.
- Chapter 6: Overall Conclusion and Future Work - This chapter concludes this thesis and provide a brief suggestion on the type of work that could be carried out in future.

In conclusion, this thesis seeks to find out the utility of machine learning in the prediction of the heart diseases through different data sources. Therefore, it is the objective of this research to use textual and ECG data from UCI repository and MIT-BIH Arrhythmia dataset, respectively, to establish a comprehensive and robust model which would enhance and refine the heart disease prediction to promote the overall patient treatment and outcome.

Chapter 2

Literature Survey

This chapter examines the existing literature on the application of machine learning (ML) and deep learning (DL) algorithms in heart disease prediction. This includes exploring the various approaches, techniques and outcomes while applying ML models in enhancing the quality of patient healthcare and the accuracy of diagnosis. This chapter is divided into two sections – related work using machine learning models and work done using deep learning models.

2.1 Work done using Machine learning algorithms

The existing work using conventional machine learning methods is covered in this section. It examines approaches like k-nearest neighbours, decision tree, random forest, support vector machines, and many more and shows how useful they are for interpreting clinical data.

Atehortua et al. [5] worked on UK BioBank data and used the XG Boost algorithm for cardiovascular disease and type 2 diabetes detection. They used SHAP to identify key features and used ROC-AUC metrics for evaluation. Zhang et al. [6] used SMOTE to address class imbalance and then combined 4 ML models, namely, Logistic regression (LR), random forest (RF), gradient boosting (GB) and artificial neural network (ANN). Itoo and Garg [7] developed a Stacking CV Classifier, constituting three ML models: KNN, NB, and LR. Whereas, Sulthana et al. [9] assessed the efficiency of NB, DT, and NB with k-means clustering. Gaikwad et al. [10] used UCI data to assess the efficiency of various ML models, such as RF, SVM, DT, GB, and LR while Jadhav et al. [11] assessed performance of machine learning models on Cleveland dataset.

Subathra et al. [4] proposed an efficient methodology, named, Bolstered Swarm Integrated Ensemble Learning, where they utilized Linear Interpolation Normalization (LIN) for preprocessing. Feature selection was performed using Bolstered-up Beetle Swarm Optimization and final classification using Weighted Ensemble Classification (WEC). Additionally, parameter tuning was done using Red Colobuses Monkey Optimization. Meanwhile, Kapila et al. [8] introduced an ensemble model, Quine McCluskey Binary Classifier (QMBC), combined with ANOVA and Chi-square to eliminate redundant or irrelevant features along with PCA for dimensionality reduction. This led to an accuracy of 98.36%. Yang et al. [12] introduced the OPTUNA framework, an advanced hyperparameter optimization

framework, with a combination of optimized LightGBM classifiers to form a model known as HY_OptGBM. With this method, they got an area under the curve (AUC) of 97.8%.

A clustered genetic algorithm approach for heart disease prediction was found by Vijaya et al. [13]. Islam et al. [14] employed PCA for dimensionality reduction, and applied a hybrid genetic algorithm via K means to achieve final clustering. In order to deal with class imbalance, Abdellatif et al. [15] used SMOTE. They then utilized Hyperband Algorithm for final prediction. Sugendran and Sujatha [21] suggested an Genetic Algorithm based Fuzzy Updating SVM for detecting heart diseases. Nanda Kumar and Narayan [23] proposed a different approach to detect heart disease. They used a hamming distance-based feature selection method and a combination of cuckoo search algorithm along with a deep belief network to find correct predictions and got an accuracy ranging between 89-91% for different datasets. Bertsimas et al. [36] trained the model using XGBoost algorithm and used OPTUNA optimization framework to fine-tune its parameters. They used SHAP (Shapley Additive Explanations), to determine features that had the highest explanatory power in the data.

Zhao and Li [22] used ECG data from Yunnan Province, China. They used genetic algorithm (GA) to determine combinations of 12 different base classifiers and then used a stacked ensemble model to diagnose ECGs. Zhang et al. [28] worked on rejecting ECGs based on data and model uncertainty with the use of the Monte Carlo dropout strategy. Liu et al. [29] discovered the Random Horizontal Flip classification method. After kernel-based pruning, they used learning rate decay-based finetuning method for ECG classification.

2.2 Work done using Deep learning algorithms

On the other hand, this subsection discusses research that has used deep learning methods to examine complicated datasets.

Dileep et al. [3] worked on the UCI dataset and used Cluster-based Bidirectional Long Short-Term Memory, and compared its approach with traditional models. Sharma and Parmar [16] used a hyperparameter optimization technique known as ‘Talos’ with the combination of the Keras library. Meanwhile, the optimization of CNN using a distance-based cat swarm optimization was the goal of Chamundeshwari et al. [17]. They utilized Field II program for removing speckle noise from ECG images along with the hybrid pattern extraction technique. Jain et al. [18] evolved the motivation for optimization. They employed Levy Flight CNN along with Sunflower Optimization Algorithm. This approach lowered the loss function of CNN and yielded an accuracy of 95.74%.

Hussain et al. [19] suggested a 1D CNN approach to predict cardiac disease. An outstanding accuracy was achieved using a dropout method. A feature selection method using genetic algorithms in combination with an ensemble deep neural network framework was introduced by K. Verma et al. [20].

The model was improved using the Adam optimizer. Yao et al. [30] used multi-scale architecture with transformer encoder to extract hidden features, making it a superior model.

Karri et al. [31] employed a different strategy for arrhythmia classification. They detected P and T waves with the help of Delta-sigma modulation and found R peaks using Discrete wavelet transform. After pre-processing and feature extraction, they used LSTM (RNN) to classify different arrhythmias. This approach resulted in a remarkable accuracy of 99.64%. Tiwari et al. [32] combined CNN and CNN-LSTM to make an ensemble framework. CNN layers were used for extracting important features which are then fed into the LSTM model.

Korurek et al. [33] proposed Ant Colony Optimization (ACO) for arrhythmia clustering. They used a lowpass filter to remove noise and a median filter for baseline estimation. The approach used by Zhang et al. [34] has two stages, namely: a feature search stage and a one-versus-one (OvO) features ranking stage. These stages are encapsulated within a binary classifier that uses OvO-rule SVM. Bouny [35] et al. used 1D-CNN model along with Stationary Wavelet Transform to extract discriminative features. Whereas in [37], the authors introduced the concept of multi-instance learning for processing ECG signals. They incorporated cross-modal information to help in the integration of instances, thereby enhancing the overall performance.

Suhail et al. [38] used discrete wavelet transform for noise removal in ECG signal, and the Nonlinear Vector Decomposed Neural Network method for predicting heart disease and achieved an accuracy of 90.67%. This work was enhanced by Ozbay Yuksel [39], who first utilized complex DWT to extract features and later used a complex-valued artificial neural network to classify arrhythmias. This resulted in an accuracy of 99.8%.

In [40], authors used a hybrid model named: CNN-LSTM. They employed a 5-fold cross-validation method and 12 convolutional layers that yielded an accuracy of 97%. This approach was enhanced by Alamatsaz [41], who worked on two datasets. Their pre-processing includes using a median filter to remove noise and down-sampling the signals from the MIT-BIH dataset to achieve the same frequency for both datasets. They then used the CNN-LSTM model and got an accuracy of 98.24%. Begum et al. [42] compared 2 models – 1D CNN and 1D CNN-LSTM model. Both these models resulted in impressive results. In [44], the authors added some random noise in the data to and oversampled it. They used a convolutional neural network (CNN). Meanwhile, Isin et al. [45] utilized ECG records 100,118, and 217 for training and 101, 107, and 231 for the testing. Pre-processing includes mean removal, high-pass filter, and moving average filter to remove noise. They combined a deep learning feature extractor with a conventional backpropagation neural network and got a success rate of 98.51%.

Table 2.1 shows a summary of work done by authors on textual and signal data.

Table 2. 1 Summary of Existing literature on Heart Disease Prediction using ML techniques

Authors	Datasets	Models used	Performance Metrics	Data Type
Yang, J. and Guan, J. (2022)	Pathological data (private)	Smote XGBoost algorithm	Accuracy, Precision, Recall, F1-Score, AUC	Textual
Dileep et al. (2022)	UCI heart disease dataset and real time dataset	Cluster-based Bi-LSTM	Accuracy, Sensitivity, Recall, F1 score	Textual
Subathra et al. (2024)	Cleveland, Statlog, and comprehensive datasets	Bolstered Swarm Integrated Ensemble Learning	Accuracy, Precision, Specificity, Sensitivity, F1-Score	Textual
Yang et al. (2023)	Framingham Heart Institute (FHS)	Light GBM, focal loss function, OPTUNA	Sensitivity, Recall, F-Score, Accuracy, Specificity, Precision	Textual
N.N. Itoo and V. Garg (2022)	Cleveland Clinic Foundation (CVF)	Ensemble model: StackingCVClassifier	Prediction, Accuracy,	Textual
Kapila et al. (2023)	Cleveland, CVD, HD dataset	Quine McCluskey Binary Classifier (QMBC) with Chi-square, Anova and PCA	Accuracy, Recall, F1-Score, Precision, Specificity,	Textual
J. Vijaya (2023)	UCI heart disease data set	Clustered Genetic Algorithm approach (CGA)	Precision, Accuracy, Error Rate, F-Score, Recall	Textual
Prerna et al. (2023)	Cleveland heart disease dataset	Hybrid Classifier in the Cloud Environment	Precision, Recall, Accuracy, F-Measure	Textual
Atehortua et al. (2023)	UK BioBank	XGBoost	ROC AUC	Textual
Zhang et al. (2022)	Henan Rural Cohort study, Dongfeng-Tongji (DFTJ)	4 ML models: LR, ANN, RF, GB	AUC and Brier Score, Sensitivity, Specificity, PPV, NPV, LR, BACC, and Youden Index	Textual
Islam et al. (2020)	UCI heart disease dataset	Hybrid Genetic Algorithm with k means + Principal Component Analysis	Accuracy	Textual

Sharma and Parmar (2023)	Heart Disease UCI dataset	Deep Neural Network (model "Optimized DNN using Talos")	Accuracy	Textual
Hussain et al. (2021)	Cleveland dataset	1-D CNN	Accuracy, Recall, F1-Score, Precision	Textual
Bouny et al. (2020)	MIT-BIH Arrhythmia database	1D CNN + SWT	Accuracy	ECG
Suhail and Razak (2022)	UCI and physio-net data repositories	DWT + Nonlinear Vector Decomposed Neural Network	Accuracy, Specificity, Sensitivity	ECG
Liu et al. (2024)	MIT-BIH	Random Horizontal Flip (RHF)-based classification	Accuracy, Precision, Recall, Specificity	ECG
Tiwari et al. (2023)	MIT BIH AR DB	CNN + CNN LSTM	Precision, Sensitivity, F-Score, Accuracy	ECG
Karri et al. (2023)	MIT-BIH dataset, QT dataset	DSM, DWT, LSTM	Accuracy, Positive Predictivity, Sensitivity, F1-Score	ECG
Zhou et al. (2024)	MIT-BIH-AR	FCBA+CNN	Accuracy, Specificity, Sensitivity	ECG
Sowmya and Jose (2022)	MIT BIH arrhythmia	CNN-LSTM	Accuracy, Precision, Recall, Specificity	ECG
Alamatsaz et al. (2024)	long-term AF database and MIT-BIH arrhythmia	CNN-LSTM	Accuracy	ECG
Isin and Ozdalili (2017)	MIT BIH arrhythmia	Back propagation neural network	Accuracy	ECG
Zhand et al. (2014)	MIT BIH arrhythmia	SVM + one-versus-one (OvO)	Sensitivity	ECG
Begum et al. (2023)	MIT BIH arrhythmia	1D CNN, 1D CNN + LSTM	Accuracy, Precision, F1-Score	ECG

Chen et al. (2023)	MIT BIH arrhythmia	cross-modal multiscale multi-instance learning approach	F1 Score, AUC, Recall	ECG
Rahul and Sharma (2023)	AFDB, CUDB, VFDB	SWT + 1D CNN + Bi LSTM	Accuracy	ECG

Discussion

As presented in the existed literature, UCI and MIT-BIH arrhythmia datasets have been the major focus of the most of the research. As previously discussed, recall, accuracy, and precision are the frequently applied performance measures for evaluating the research work. CNN is the most popular deep learning model.

However, in most heart disease prediction studies, feature selection, outlier treatment, and hyperparameter tuning get less attention. Thus, this thesis attempts to fill these gaps by investigating the present models and developing a more refined approach to increase the accuracy of the prognosis.

Chapter 3

Preliminaries

This chapter lays the foundation of methods used to convert unprocessed data into formats appropriate for model training and assessment in the prediction of heart disease. Preprocessing, feature extraction, feature selection, and model training and evaluation techniques are all covered in detail. Additionally, it also provies a thorough explanation of the nature-inspired algorithms employed along with machine learning (ML) and deep learning (DL) models employed throughout this thesis.

In the context of predicting diseases using machine learning, there are several steps like:

1. Pre-processing - This involves eliminating noise from data, along with standardizing and normalizing data to enhance model accuracy.
2. Feature extraction – This step involves extracting features from the data to capture relevant information.
3. Feature Selection – Here, the most essential features are chosen to lower the dimensionality of data and enhance computational speed.
4. Model Construction – Selecting the appropriate model depending on nature, size, and characteristics of data.
5. Evaluation – Performance measures such as accuracy, f1-score, recall, specificity, and precision are used to assess the behaviour of model.

Each of these steps is elaborated below.

3.1 Pre-processing

Data preprocessing is an initial step in preparing raw data for predictive modelling. It includes handling noise, outliers, and duplicates, which can negatively impact model performance. This subsection discusses the need of preprocessing to improve data quality, reduce noise, and ensure consistency and dependability in subsequent studies.

3.1.1 Types of Noise and Artifacts

ECG Signal Noise

ECG signals may be affected by several types of noise and artifacts that can lower signal quality and decrease model performance. Therefore, noise removal is crucial before analyzing ECG signals.

Types of noise are:

1. Powerline Interference - Electrical interference at 50 or 60 Hz
2. Baseline Wander – Low-frequency noise resulting from breathing or body movement.
3. Electrode Contact Noise - Noise due to poor skin-electrode contact.
4. Electrosurgical Noise – Electrical disturbance from nearby equipment.
5. Random Noise - Unpredictable amplitude variations.
6. Muscle Contractions - Noise caused by non-heart muscle activity.

Textual Data Noise

In the case of textual data, there may be some kind of noise that must be removed.

1. Stop words: Removing stop words like ‘and’, and ‘the’ is mandatory for correct analysis
2. Spelling Errors: Misspelled words can lead to the wrong interpretation of data.
3. Special Characters: Unnecessary symbols like ‘@’, and ‘$’ do not contribute in any way.
4. Case Sensitivity: Wrong capitalizations can cause inconsistencies in data.

3.1.2 Data Cleaning Techniques

Dealing Null values

The most common issue in both ECG and textual data is missing values. Two main strategies to deal with null values are:

1. Removal: If a few rows/columns have null values, they can be deleted.
2. Imputation: Imputation means filling null values using mean (in case of numerical data), median (in case of outliers) or mode (in case of categorical data).

Removing Duplicates

Repeated measurements in ECG may cause duplicate entries in ECG data, while identical text entries can cause redundancy in case of textual data. Methods to remove duplicates are:

1. Exact Matching: Identify same records and delete them.
2. Near-Duplicates: Use ML techniques to identify nearly same records and delete them.

3.1.3 Data Transformation

Standardization and Normalization

To make data consistent on the same scale, it is important to normalize and standardize data.

1. Standardization: converting data into a form where the mean is zero and the standard deviation is one.
2. Normalization: It is the process of scaling the value of features to a consistent scale, usually between 0 and 1. The main aim is to prevent features from dominating the model training because of their large values. Common methods for normalizing are Min-max normalization and z-score normalization [41].

Specific Filtering for ECG Data

Filtering in ECG signals to enhance data quality includes:

1. High-pass Filter - Removes baseline wander, which is set around 0.5 Hz.
2. Low-pass Filter - Eliminates high-frequency noise like muscular interference (set around 100 Hz).
3. Bandpass Filter - Retains a specific frequency range, isolating the QRS complex.
4. Notch Filter - removes a narrow range by combining high-pass and low-pass filters (for instance, 50 or 60 Hz for powerline interference).
5. Median Filter - Suppresses abrupt fluctuations by substituting data points with the median value.
6. Moving Average Filter - Softens the signal by averaging data points inside a window, lowering high-frequency noise.

Text Pre-processing Methods

To convert raw data into suitable form, commonly used methods are:

1. Tokenization- It means splitting data into individual words.
2. Lowercasing – To ensure uniformity, the whole data is converted into a lowercase.
3. Stemming/Lemmatization- It includes the reduction of words into their base form (e.g., playing can be written as play).

3.2 Feature Extraction

Extracting features from both textual data ECG signals is the crucial step in signal processing [47]. Raw data is transformed into a set of important features for our model.

3.2.1 ECG Data Feature Extraction

PQRST Complex

P, QRS, and T complexes provide information regarding cardiac problems. Frequently used methods for detecting QRS complex include Hamilton-Tompkin, Pan-Tompkin, and Fuzzy logic-based methods [47].

1. P Wave: Reflects atrial depolarization with a duration of 0.08-0.1 seconds.
2. QRS Complex: Depicts ventricular depolarization and comprises three waves -Q, R, and S. Fluctuations in the height of the R wave indicate abnormalities.
3. T Wave: Represents ventricular repolarization, typically lasting 0.16 seconds.
4. U Wave: Minor deviation with unclear significance.

Figure 3.1 represents a general ECG signal.

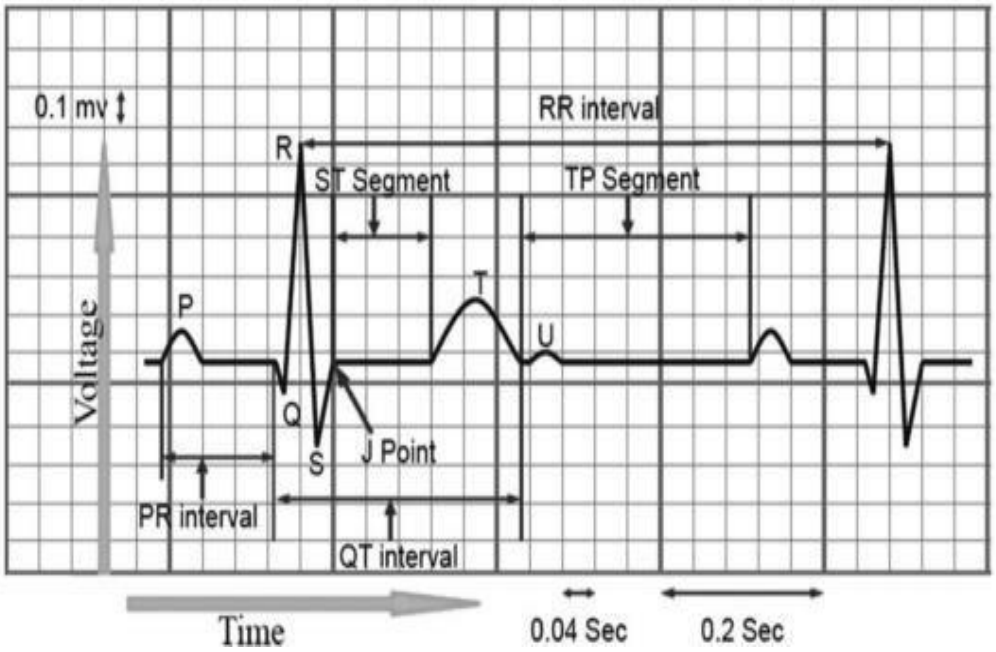


Figure 3.1 Standard Fiducial points of ECG signal [43]

Statistical Features

Quantitative insights are obtained using statistical features like mean, standard deviation, skewness, entropy, and correlation coefficients [46].

1. **Mean:** Average value of the signal.
2. **Standard Deviation:** Measure of signal variability.
3. **Skewness:** Asymmetry in the distribution of signal.
4. **Entropy:** Measures the complexity of the signal.
5. **Correlation Coefficients:** They provide relationships between different signal segments.

Morphological Features

Morphological features contain unique waveform patterns, important for identifying various health conditions. These features include the signal's shape, segment, duration, slope, and amplitude for identifying abnormalities [45].

3.2.2 Textual Data Feature Extraction

In the case of textual data, feature extraction means converting text data into numerical form so that the model can process it easily. It involves methods like:

Term Frequency- Inverse Document Frequency

TF-IDF adjusts the word frequencies based on their importance. Weights of common words are reduced, while rare significant words are highlighted.

Bag of Words

Using this approach, text is represented in the form of the frequency of words in a document. Thus, a vector is created for each sample of text.

3.3 Feature Selection

In ECG analysis, feature selection is important for reducing the dimensionality of large datasets. This section discusses three primary methods, namely, filter method, wrapper method, and embedded method, for feature selection.

3.3.1 Filter method

This method gives scores to all the features and is independent of the learning model [8]. Scoring methods include:

1. **Chi-Squared Test:** It calculates the independence between input features and target variables.
2. **Fisher Score:** It evaluates the discriminative power of features.
3. **Mutual Information:** It measures the common information between input features and the target.
4. **Correlation-Based Feature Selection (CFS):** It selects features on the basis of their correlation with the target variable.

3.3.2 Wrapper method

It is used to evaluate features using a specific search method and learning model. It is computationally more expensive [46]. Examples include:

1. **Sequential Forward Selection (SFS):** It adds one feature at a time based on improvement in performance.
2. **Recursive Feature Elimination (RFE):** It removes the least important features iteratively.
3. **Sequential Backward Selection (SBS):** It removes features one at a time based on performance deterioration.

3.3.3 Embedded Approaches

This approach incorporates feature selection into the training process of the classifier. It leverages the model's internal algorithms to determine the feature significance [46]. Examples include Lasso regression, where L1 regularization shrinks less important features.

3.4 Model Construction and Evaluation

The final step is constructing the appropriate model and evaluating its performance. This section gives summary of common models and evaluation metrics used in disease prediction.

3.4.1 Model Selection

The model selection depends on the nature of data. Common models used in ML are:

1. Linear Models: These include linear and logistic regression.
2. Tree-based models: These comprise random forests and decision trees.
3. Neural Networks: These models are useful to deal with complex datasets.

The detailed explanation is given in subsequent section.

3.4.2 Evaluation Metrics

Precise performance measures are essential for evaluating the performance of proposed method. Table 3.1 shows the evaluation metrics used in this research.

Table 3. 1 Evaluation metrics used

Measure	Estimation	Description
Accuracy	$\frac{TP + TN}{TP + TN + FN + FP}$	Gives the ratio of correctly predicted instances to total number of instances.
Specificity	$\frac{TN}{TN + FP}$	Gives the proportion of real negatives that are appropriately identified.
Recall	$\frac{TP}{TP + FN}$	Gives the percentage of real positives that are appropriately identified.
Precision	$\frac{TP}{FP + TP}$	Gives the proportion of positively identified points that are actually correct
F1 score	$\frac{2 * recall * precision}{recall + precision}$	Gives harmonic mean of recall and precision, providing a balance between the two

3.5 Optimization algorithms and Machine Learning models

This subsection gives an in-depth explanation of all the optimization algorithms (nature inspired algorithms) used for feature selection and machine learning models for final classification in this thesis. Feature selection significantly improves model performance by finding the most relevant features, thereby reducing computational complexity.

3.5.1 Nature-Inspired Algorithms

Nature-inspired algorithms are inspired from biological processes to deal with complex problems. This section explores two important nature-inspired algorithms, namely, Genetic Algorithm and Cuckoo Search Algorithm.

Genetic algorithm (GA)

Genetic algorithm (GA) is a powerful search algorithm based on principle of natural selection and is widely used in addressing machine learning and optimization problems [21]. The major objective is to refine feature subsets iteratively to reach a high-quality solution close to the optimal solution without

knowing the search space [22]. This algorithm works on the fundamental concept of “survival of the fittest”.

Steps involved in GA are:

1. Initialization: Initially, a population of individuals, known as chromosomes is generated randomly.
2. Selection: During each iteration of generations, an individual's fitness is calculated using some predefined fitness function.
3. Crossover: The Crossover helps in the mating between the above-selected individuals [21]. Genetic information between randomly selected crossover sites is exchanged to produce new offspring that comprises traits from both parents.
4. Mutation: Mutation introduces random genes in the genetic makeup of individuals, ensuring population diversity and avoiding premature convergence.
5. Replacement: Old population is replaced with new population, thus forming next generation.
6. Termination: With each passing generation, the solution approaches its optimal solution. The algorithm terminates on attaining a high-quality solution as shown in Figure 3.2, while Figure 3.3 visually represents crossover and mutation.

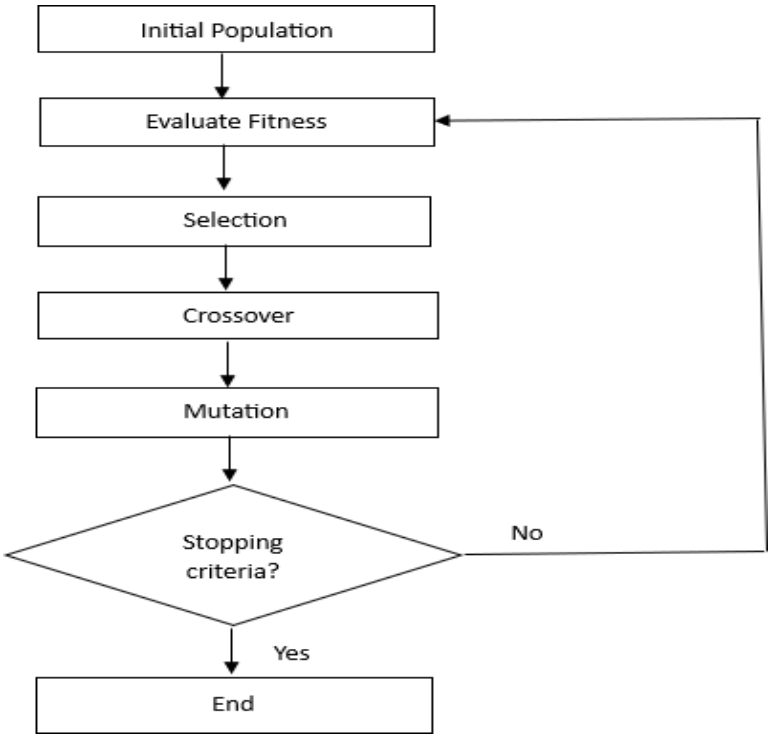


Figure 3.2 Flowchart of Genetic Algorithm

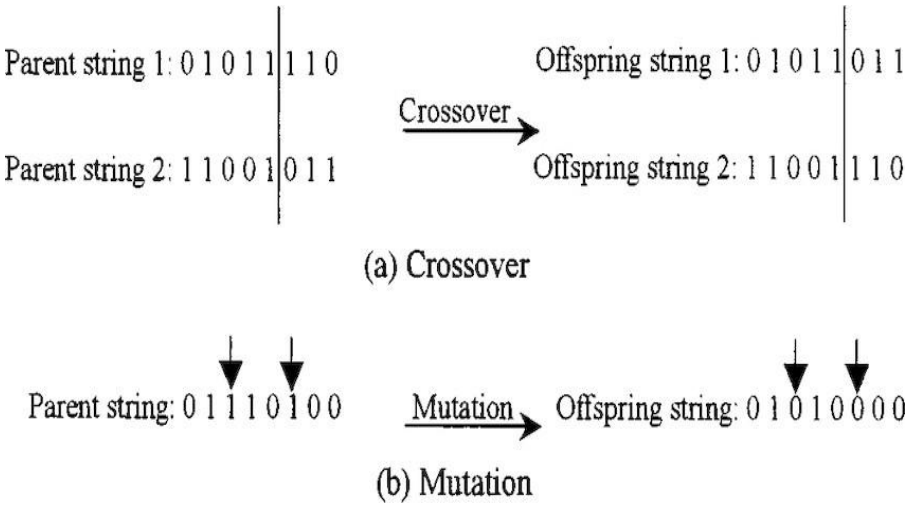


Figure 3.3 Crossover and Mutation

Cuckoo Search Algorithm (CSA)

Cuckoo Search algorithm (CSA) is a nature-inspired method widely applied in domains like speech recognition, cloud computing, and software testing [23]. It is based on the way some species of cuckoo lay their eggs in the nests of other birds, or hosts. Host birds can either reject the egg or build a new nest. Each nest represents a potential solution to the optimization problem. Fitness for each nest is calculated during each iteration and the quality of a solution is assessed.

Levy flight, a key mechanism in CS, refers to a random walk where step lengths follow Levy distribution with a heavy tail. This enables the algorithm to make long jumps across the search space thus exploring diverse regions. Steps involved in CS optimization algorithm are:

- Initialization: Initialize a fixed number of nests (n) within the solution space [24].
- Evaluating Fitness: Fitness is computed for each nest to identify the best solutions. The fitness of a newly laid cuckoo egg, representing a potential solution, is compared with that of the host egg within the nest.
- Levy flight: Nests are replaced with new ones based on the levy flight mechanism, fostering diversification.
- Egg laying: If the fitness cuckoo's egg is higher than existing eggs, nest is replaced with new solutions.
- Termination: The process terminates after reaching a predefined stopping criterion.

Figure 3.4 illustrates the basic flowchart of cuckoo search algorithm.

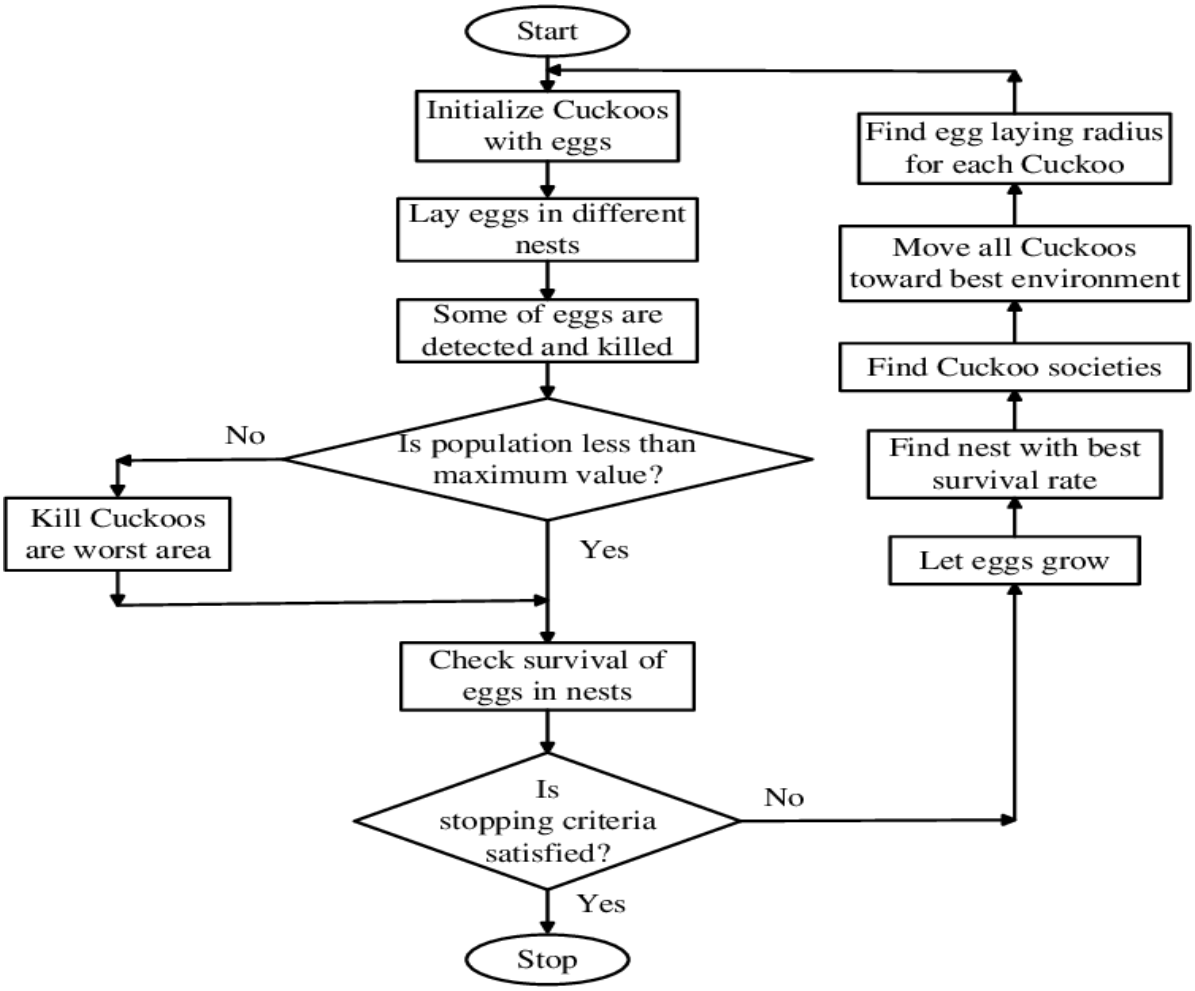


Figure 3.4 Flowchart of the Cuckoo search algorithm [54]

3.5.2 Machine Learning Algorithms

This subsection provides explanation of all the machine learning (ML) models employed in this thesis for heart disease prediction. These algorithms play an important role in analyzing complex datasets to identify intricate patterns that contribute to the disease diagnosis.

Support Vector Machines (SVM)

Support vector machine (SVM) is a type of supervised machine learning model used in regression, classification, and outlier detection tasks. Introduced in 1990, SVM works very well on high-dimensional data. It works by finding the hyperplane (decision boundary) between different class target variables.

The equation of hyperplane in SVM is:

$$y = w^T x + b \quad (1)$$

Where:

- x represents input feature
- y represents decision value
- w represents the weight vector
- b is bias term

Two main types of SVM are:

1. Hard-margin SVM– If the hyperplane separates data points of different classes without any error (misclassification), it is called hard-margin SVM.
2. Soft-margin SVM – If the data points are not perfectly separable or are misclassified, then soft-margin SVM is used.

Extreme Gradient Boosting (XGBoost)

Introduced in 2014, the XGBoost algorithm is an ensemble learning method widely popular because of its exceptional performance in classification and regression tasks, typically on large-scale datasets. XGBoost stands for 'Extreme Gradient Boosting' and is well known for its accuracy and scalability. It combines multiple weak learners iteratively to build a stronger learner.

XGBoost leverages the power of decision trees and enhances the outcome using a boosting approach. Boosting refers to the combination of weak learners to form a robust model. The steps involved in XGBoost are:

1. Initialization: The process begins with a simple prediction like the mean of the target feature.
2. Iteration: During successive iterations, a new decision tree is added to the ensemble, which predicts the error.
3. Weights: Each input feature is assigned a weight based on its importance.
4. Tree Construction: Information gain (IG) is used for effective splitting of the nodes. IG measures impurity in target feature.
5. Update: New predictions are added to previous predictions of the ensemble model, thus increasing accuracy. The process terminates after a certain number of iterations.

XGBoost involves parameters such as the number of trees, tree depth, learning rate and regularization parameters (e.g., lambda) to reduce overfitting [18]. These hyperparameters can be optimized automatically using Parzen estimation strategy. Therefore, XGBoost is a powerful model for high-dimensional datasets.

Random Forest (RF)

Random forest is an ensemble tree learning method used in both classification and regression. It employs multiple decision trees and combines their results to improve the model's performance.

A Key concept used in Random Forest is Bootstrap aggregation or Bagging.
Bagging- Instead of feeding the whole data into a decision tree, bootstrapped data is used. Bootstrap data refers to subsets of data selected randomly with replacement from the original dataset. This reduces the chances of overfitting as this randomness in data causes variations among individual trees.

Baic steps in random forest are:

1. Bootstrapping: Multiple bootstrap samples are created with replacement. Some data points may not appear at all in any of the bootstrapped data.
2. Building decision tree*:* A decision tree is constructed for each bootstrap sample with a random subset of features.
3. Aggregating Predictions: The final prediction is made by combining results from all the trees – majority voting in case of classification, and average value for regression.
4. Evaluation: Model is evaluated using performance metrics like accuracy, recall, precision.
5. Hyperparameter Tuning: When looking for best possible split, optimize factors like maximum depth, number of trees, and minimum samples for leaf node using Grid Search cross-validation.
6. Final Training: After optimizing the parameters, model is trained again and evaluation metrics is noted.

Random forest stands out as a powerful ML algorithm because of its higher accuracy.

K-Nearest Neighbor (KNN)

KNN is an instance-based learning model used for both classification and regression tasks. According to KNN, similar data points lie close to each other. The main advantage of KNN is that it is a non-parametric method, i.e., it doesn't need any assumption about data distribution. For a given data point, KNN identifies k nearest examples closest to it and assigns majority class labels. Correct selection of 'k' value is very crucial as it can impact the performance of model.
Steps involved in KNN are:

1. Select appropriate value of number of neighbors (k). Cross-validation method can be used to choose best value for 'k'
2. Select distance metrics to calculate distance between data points. Commonly used distance metrics is Minkowski distance given by equation 2.

$$d(x, y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} \quad (2)$$

where n represents total number of data points

In above formula,

if p=1, it becomes Manhattan distance and If p=2, it becomes Euclidean distance.

3. Calculate distance between given data point and all other points of training examples
4. Identify ‘k’ points that are closest to the given point
5. Out of those ‘k’ points, find the majority class and assign it to test data point.
6. For each test data, apply above steps to find class labels.

Logistic Regression (LR)

Logistic regression is a supervised ML model used for binary classification purpose. LR uses logistic function or sigmoid function, which gives a probabilistic value between 0 and 1 based on predictor variables. This sigmoid function makes an S-shape curve with a threshold value. Values above threshold are noted as 1 while below threshold are noted as 0. The logistic function is defined in equation 3.

$$h(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Naïve Bayes (NB)

Naïve Bayes is a probabilistic model used for classification tasks. Naïve Bayes is based on Bayes Theorem, which is defined as the probability of an event based on prior conditions (conditional probability) [27].

The basic assumptions of Naïve Bayes are:

1. All the features are mutually independent of each other.
2. Continuous features follow the normal distribution.
3. Discrete features follow multinomial distribution.
4. All features are considered equally important for the target variable.

Following steps are involved in Naïve Bayes:

1. Prior Probabilities

   Compute Prior probabilities for each unique value of target feature
2. Likelihoods

   Compute the likelihood of each unique value for each input feature.

3.5.3 Deep learning models

This subsection explores the DL models used in this thesis. These DL models can capture dependencies within data and extract complex hidden patterns.

Convolutional Neural Network (CNN)

CNN is a deep learning model used in healthcare sector. Although CNNs require high computing power, it is used in healthcare fields because of its accurate predictions.
After splitting that data into 80% training data and 20% testing data, the preprocessed goes through several layers of CNN.

1. Convolutional Layer
    - Convolution Operation – It captures local patterns in input data by applying filters (kernels) over input data.
    - Activation Function – Activation function, like ReLU is used to add some non-linearity into model, enabling it to capture more patterns.
2. Pooling layers
    - Max Pooling – This layer decreases feature map dimension generated by convolution layer, using a fixed size window. This layer reduces computational load.
3. Fully Connected Layers
    - Dense layer – This layer receives the flattened input received from above 2 layers and connect neurons so as to capture complex patterns.
    - Output layer – It has sigmoid activation function for final predictions.

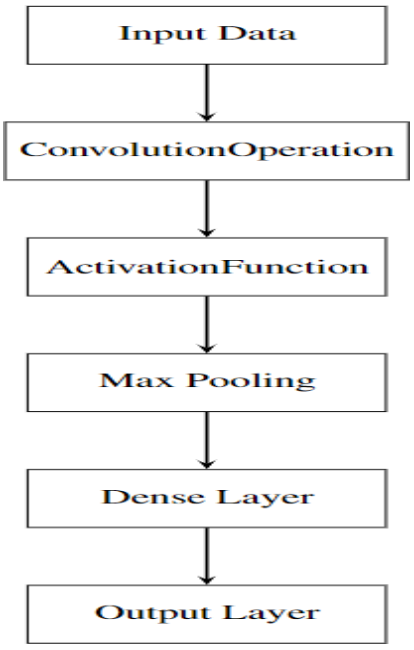


Figure 3. 5 Flowchart of CNN

Recurrent Neural Network (RNN)

RNNs are a family of neural networks intended for sequence modeling. They perform best for problems where sequence of input is crucial, like, speech recognition, and time series prediction. RNN has a hidden state, that keeps on changes with respect to previous hidden state and current input [53].

Theoretically, RNNs can easily capture long-term dependencies, but in practice, they face difficulty due to expanding and vanishing gradient problems.

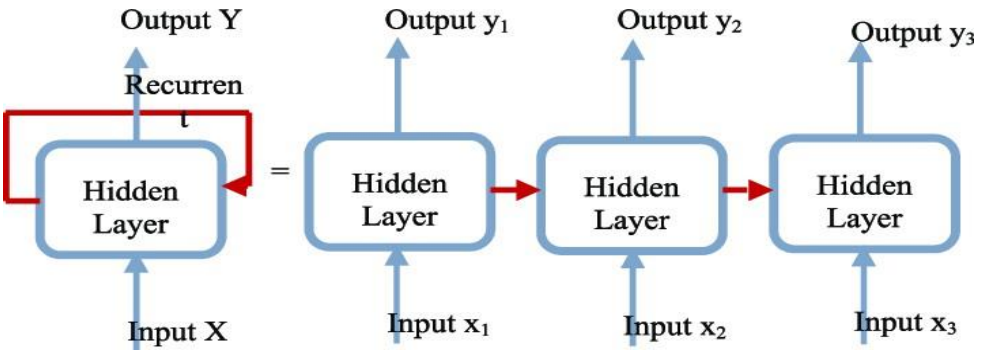


Figure 3. 6 Architecture of RNN [53]

Long Short-Term Memory (LSTM)

LSTM is a type of RNN introduced in 1997 to deal with sequential data. LSTM takes three inputs – current input, previous cell state, and previous hidden state. LSTM can capture long-term dependencies effectively by passing the input through three gates: forget, input and output gate, each having sigmoid activation function [52].

1. Forget gate – This gate tells which information is to be discarded from the cell state.
2. Input Gate – It decides which information should be added to the cell state.
3. Cell State Update - In this step, cell state is updated on basis of new candidate value generated using tanh layer and previous two gates.
4. Output Gate – It informs what part from the cell state will be considered as output.

This process results in two outputs, namely, hidden state and next cell state.

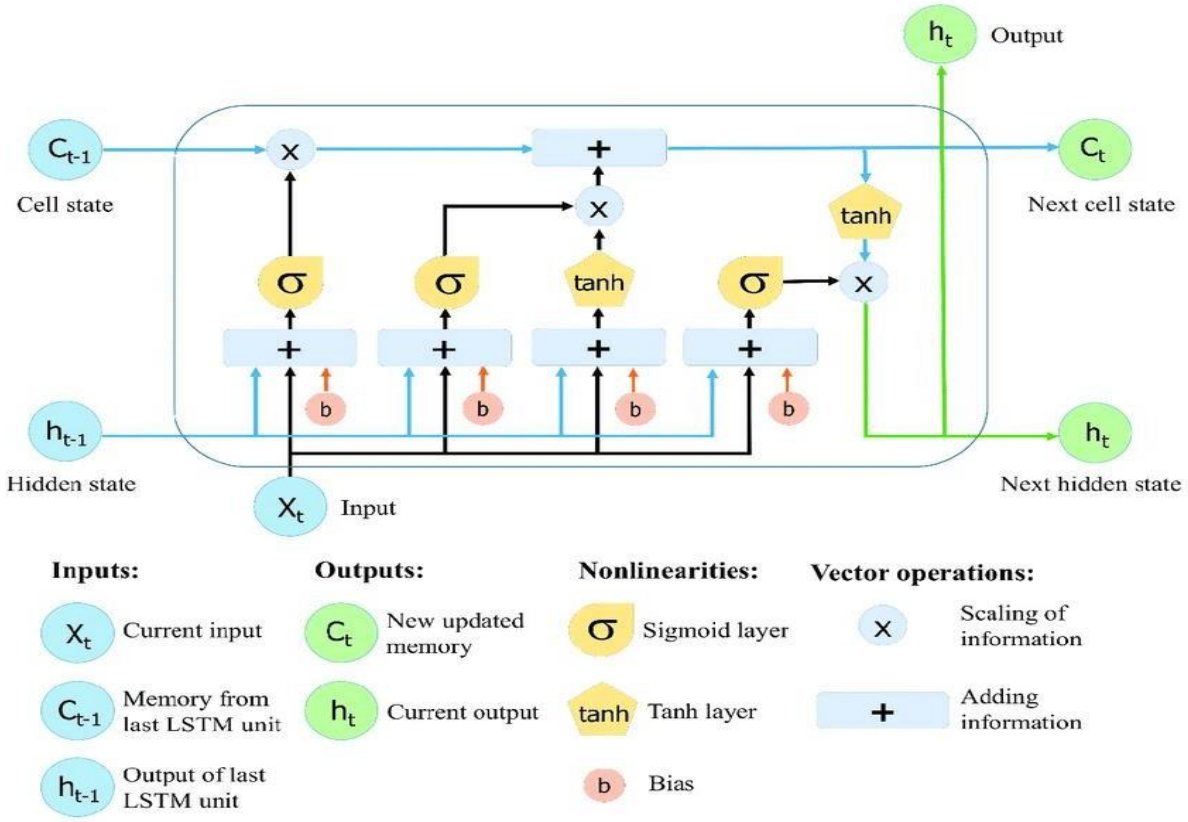


Figure 3. 7 Working of LSTM [52]

Bidirectional Long Short-Term Memory (Bi-LSTM)

Bi-LSTMs are extended version of LSTM that capture information from past and future [51]. Their architecture has mainly two layers- forward and backward layer. Input is processed from both directions while outputs are combined from both LSTM layers.

1. Input Layer- The input sequence of feature vectors enters the Bi-LSTM network.
2. Forward LSTM layer– This layer deals with processing of sequence from beginning to end (left to right). Just like LSTM, this layer uses forget gate, input gate and then update cell state on basis of results from above 2 gates. Using output gate, it passes filtered version of cell the next hidden state.
3. Backward LSTM – This layer deals with processing of sequence from end to beginning (right to left). This layer work similarly as forward LSTM layer and enables the network to capture future context for each input.
4. Combining outputs - This layer concatenates the outputs from above two layers.
5. Output Layer- For each time step, this layer receives the concatenated information from both forward and backward layers.

Bi-LSTMs are majorly used for speech recognition, and machine translation.

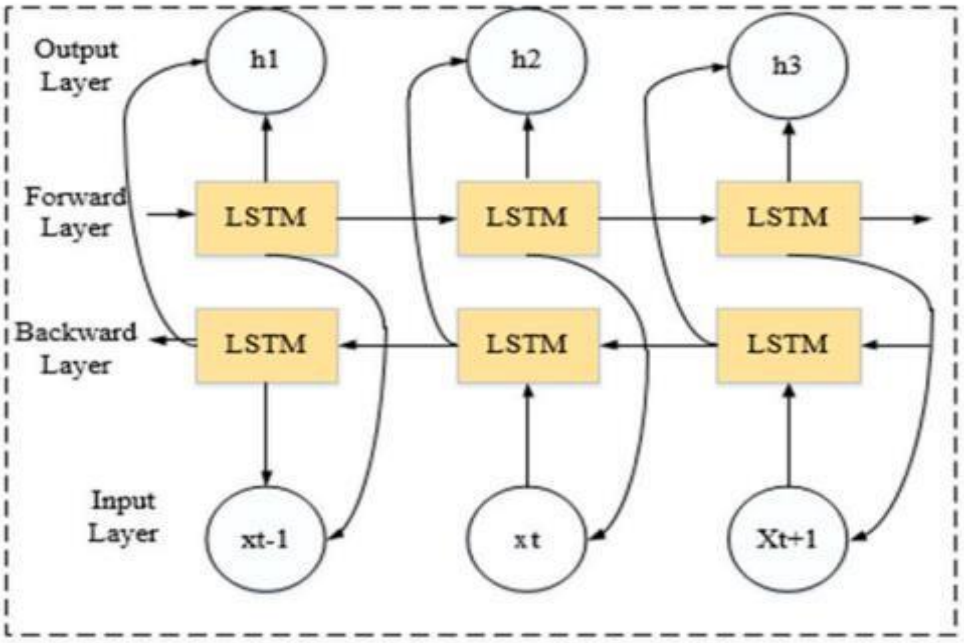
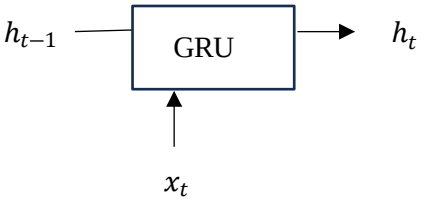


Figure 3. 8 Working of Bi-LSTM [51]

Grated Recurrent Unit (GRU)

GRUs, an alternative to LSTMs, use fewer gates as compared to LSTM. GRU doesn't have separate cell state unlike LSTM. Rather it only has input and hidden state.



GRU takes an input and previous hidden state and outputs a new hidden state. Architecture of GRU has two gates namely, reset gate and update gate.

1. Reset Gate – It controls the amount of previous hidden state that has to be ignored.
2. Update Gate - It informs about the part of previous hidden state that has to be carried forward.
3. New Memory Content – This step creates new memory content.
4. Final Memory – It gives the final memory at current time step.

Chapter 4

Comparative analysis of existing models

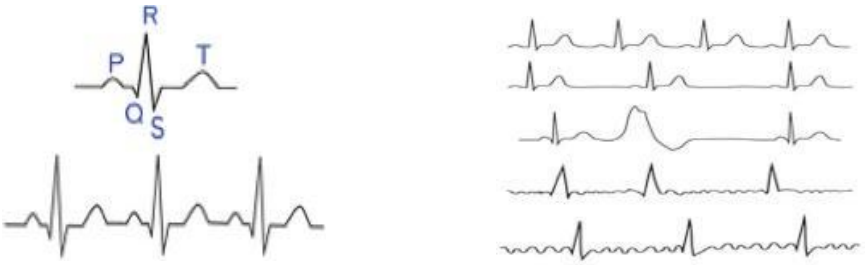
This chapter uses MIT-BIH Arrhythmia Dataset to conduct a thorough comparative analysis of existing models for heart disease prediction. It has four main sections, experimental setup, methodology, results and discussion, and major findings and conclusions. By examining a range of machine learning (ML), deep learning (DL), and hybrid models, this analysis aims to identify the most effective approaches for accurately diagnosing cardiac abnormalities.

4.1 Experimental Setup

The experimental setup includes a number of components that are required for carrying out a thorough examination of the proposed algorithms for the prediction of heart disease. The Google Compute Engine handled the entire heart disease prediction approach that was conducted using Python 3 on Google Colab. The dataset and evaluation metrics utilized are described in this section.

Dataset Description

Our study uses MIT-BIH Arrhythmia Database. The data constitutes two-channel ECG recordings. The roughly 30-minute recordings, which were taken from 47 people—25 men and 22 women—provide a thorough picture of a various arrhythmias. The recordings in the collection have two channels: V1 as the second channel and a modified limb lead II (MLII) as the first, both with a 360 sampling rate. Five categories of arrhythmias are included in this dataset: Q (Unknown beat), F (Fusion), V (Ventricular ectopic beat), and N (Normal). We simplified the five original classes into a binary classification, distinguishing between arrhythmia and regular rhythm. Figure 4.1 illustrates the difference between normal and abnormal ECG signals, while Figure 4.2 displays the location of the 12-lead electrodes.



Normal ECG signal Abnormal ECG signal

Figure 4.1 Normal vs Abnormal rhythm [23]

Figure 4.2 12-Lead electrode [23]

4.2 Methodology

This section describes framework for various existing ML, DL, and hybrid algorithms for heart disease prediction. It elucidates the different preprocessing and feature selection method on each model.

Machine Learning models

We implemented different combinations of ML models along with different filtration techniques and came out with five distinct models as follows:

M-1 K-nearest neighbour (KNN) with Select from model (SFM)
The data was first pre-processed using high-pass filter and then normalized through min-max normalization. To handle class imbalance, random over-sampling technique was used. Significant features were identified from this resampled data using Select from model (SFM) method, an embedded feature selection method that combines logistic regression with L1 regularization [49]. This method selects important features based on weights assigned by the training model. These features are then fed into KNN neighbour algorithm.

M-2 Random Forest (RF) with Select from model
This method starts with a high-pass filter followed by normalization. It employs SMOTE to generate synthetic samples for the minority class. The SFM method is used to reduce overfitting. This processed data is then fed into a random forest for final classification.

M-3 Random Forest (RF) with Backward Sequential feature selection (SFS)

Here, data is pre-processed using high-pass filtering, normalizing and balancing using Smote-Tomek [47]. The backward Sequential feature selection (SFS) method [47], a wrapper method, is employed to select key features. Features are removed one by one from the current set of attributes based on performance. Random forest model is then trained on training data.

M-4 Random Forest (RF) with Backward Sequential feature selection (SFS) and smote

This model is similar to M-3 but uses the SMOTE balancing technique instead of Smote-Tomek, resulting in a slight accuracy improvement to 98.13%.

M-5 Ensemble model

In this method, we use the SMOTE balancing technique, to address class imbalance. Five ML algorithms, namely, DT, KNN, RF, GB, and NB are trained on the resampled data. Predictions were made on unseen data by each model, and the results were averaged. Performance metrics were calculated using a threshold of 0.5.

Deep Learning Models

This subsection presents the explanation of various deep learning algorithms combined with various sampling and filtering techniques.

M-6 Autoencoder

After first undergoing pre-processing with a low-pass filter, data is normalized. To handle class imbalance, a random over-sampling technique is used. The Autoencoder is then trained using this resampled data to decrease dimensionality. A sequential model comprises a dense layer with 64 neurons and an output layer with a single neuron.

M-7 Long Short-Term Memory (LSTM)

This method employs random over-sampling to balance the data and LSTM for final prediction. LSTM stores and retrieves data via memory cells. Three gates are used in this algorithm, namely, input, forget, and output. During training, the model learns to utilize and update its memory cells.

M-8 U-NET

Once data has been balanced using the Smote-ENN technique, U-net, a deep learning algorithm, is used for training and testing. U-Net has a U-shaped design that is comprised of an encoder and a decoder. Skip connections, which are located between the encoder and decoder, are used to record fine-grained details.

M-9 Convolutional Recurrent Neural Network (CRNN)

Recurrent and convolutional layers are combined to form CRNN. While recurrent layers like LSTM help in understanding dependencies among various characteristics, convolutional layers help capture spatial elements within ECG data. The last layer for the prediction of binary heart disease is a dense classification layer. After employing random-over sampling, the resampled data is used to train this model.

M-10 Gated Recurrent Unit (GRU)

GRU is used to find transient patterns in ECG data. Just like CRNN, a dense layer with a sigmoid activation function is added for final classification.

M-11 Ensemble of CNN and Bi-LSTM

CNN + Bi LSTM combines the advantages of both CNN and Bidirectional LSTM. CNN helps to capture local features, while Bi-LSTM captures long and short-term dependencies. This combination enables the model to predict heart disease effectively.

Hybrid Models

Hybrid models combine multiple ML and DL models to leverage the advantage of individual model and enhance overall performance.

M-12 Random Forest with Convolutional Neural Network (RF + CNN ensemble)

Here, data was normalized using a min-max scaler followed by using a combination of random forest and convolutional neural network for classification. The random forest has 100 trees while the CNN structure has one convolutional layer, one max-pooling layer, one flatten layer, one dense layer with 50 units, and an output layer having sigmoid function. The averaging method is used to combine the predictions.

M-13 Random Forest with Long Short-Term Memory (RF + LSTM ensemble)

Long short-term memory (LSTM) architecture with a single LSTM layer with 50 units and an output layer with a sigmoid function has been used along with random forest. The averaging method is used to combine their predictions.

M-14 RF + CNN+ LSTM ensemble

This method integrates predictions of RF, CNN, and an LSTM in order to predict heart disease. A 98.42% accuracy rate is obtained.

4.3 Results and Discussions

This section discusses the results and outcomes obtained by employing various models for predicting heart diseases. This is done using quantitative evaluation metrics like accuracy, precision, recall and specificity.

Machine learning models

The results of the traditional ML techniques are shown in Table 4.1 and Figure 4.3.

Table 4.1 Performance Comparison of various Machine Learning Algorithms

Models	Accuracy	Precision	Recall	Specificity
M- 1	98.39%	90.33%	94.74%	98.81%
M- 2	98.21%	89.77%	93.6%	98.75%
M- 3	98.12%	89.18%	93.41%	98.67%
M- 4	98.13%	89.15%	93.55%	98.67%
M- 5	**98.67%**	**91.37%**	**96.44%**	**98.94%**

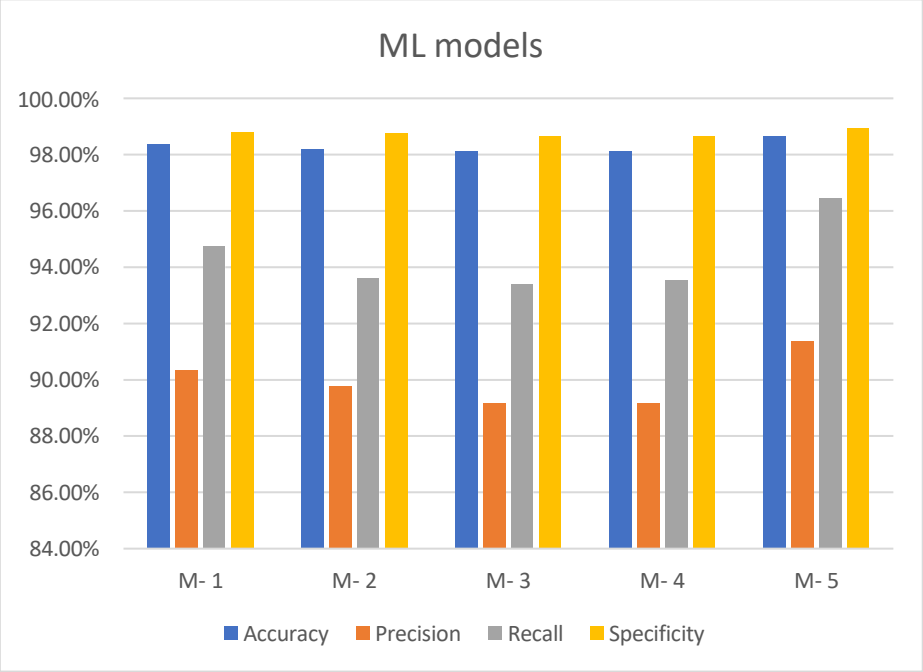


Figure 4.3 Graphical representation of the evaluation metrics for ML models

- Table 4.1 and Figure 4.3 show that, among the five machine learning models (M-1 to M-5), M-5 has the best accuracy (98.67%), followed by M-1, M-2, M-3, and M-4, which are nearly all at 98%.

- M-5 scores the highest in precision (91.37%), closely followed by M-1 (90.33%). M-2, M-3, and M-4, on the other hand, have nearly identical accuracy of roughly 89%.
- In addition, M-5 attains the maximum recall rate of 96.44%, demonstrating its efficacy in detecting positive instances. At 93.41%, M-3 has the lowest recall.
- M-5 scores the highest (98.94%) in terms of specificity, followed by M-1 (98.81%), M-2 (98.75%), M-3, and M-4 (98.67% apiece).
- M-5 is the best model all around.

Deep Learning Models

The comparative results of deep learning models are presented in Table 4.2 and Figure 4.4.

Table 4.2 Comparative analysis of various deep learning algorithms

Models	Accuracy	Precision	Recall	Specificity
M-6	95.16%	**93.57%**	**97.03%**	93.28%
M-7	98.54%	90.17%	96.54%	98.76%
M-8	98.54%	90.45%	96.20%	**99.29%**
M-9	98.58%	91.34%	95.54%	98.94%
M-10	**98.69%**	91.91%	95.92%	99.01%
M-11	98.11%	90.87%	91.18%	98.29%

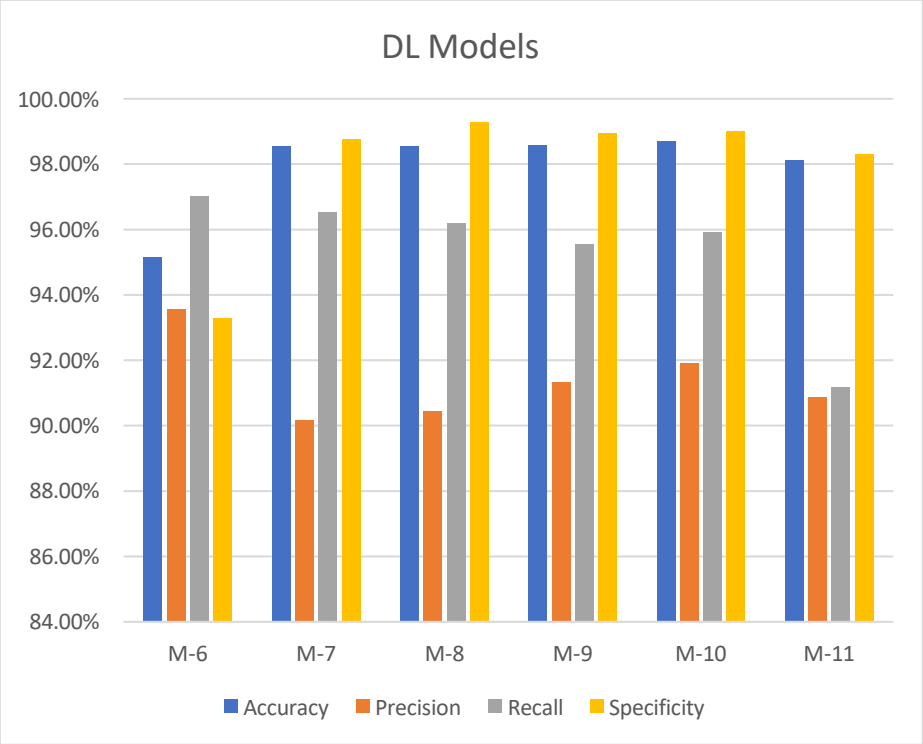


Figure 4.4 Graphical representation of the evaluation metrics for DL models

- Out of six deep learning models (*M-6* to *M-11*), in terms of accuracy, M-10 has the highest score of 98.69%. M-9 (98.58%), M-7, and M-8 all have similar accuracy scores of 98.54%. At 95.16%, M-6 has the lowest accuracy.
- M-6 has the highest precision rate (93.57%), followed by M-10 (91.91%), M-9 (91.34%), M-11 (90.87%), and M-8 (90.45%). The precision of M-7 is the lowest, at 90.17%.
- With a recall value of 97.03%, M-6 has the highest recall, indicating its capacity to identify positive instances. At 91.18%, M-11 has the lowest recall. Recall percentages for M-7, M-8, M-9, and M-10 are 96.54%, 96.20%, 95.54%, and 95.92%, respectively.
- M-8 performs best in terms of specificity, scoring 99.29%, closely followed by M-10, which scores 99.01%. M-6, on the other hand, has the lowest specificity (93.28%). The specificities of M-7, M-9, and M-11 range from 98 to 99%.
- M-6 is the most reliable model because to its higher recall.

Hybrid Models

The results of the proposed hybrid models are shown in Table 4.3 and Figure 4.5.

Table 4.3 Comparative analysis of various hybrid algorithms

Models	Accuracy	Precision	Recall	Specificity
M-12	**99.13%**	**98.35%**	**93.26%**	**99.81%**
M-13	**99.13%**	**98.35%**	**93.26%**	**99.81%**
M-14	98.42%	97.15%	87.52%	99.70%

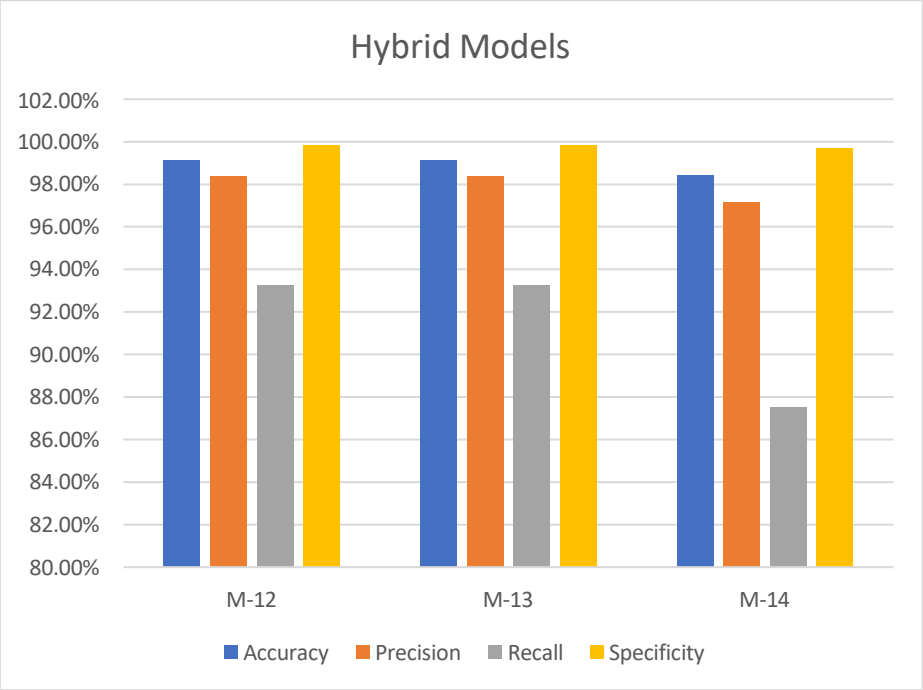


Figure 4.5 Graphical representation of the evaluation metrics for hybrid models

Regarding the hybrid models (M-12 to M-14), Table 4.3 and Figure 4.5 findings show that models M-12 and M-13 show exceptional performance with the highest values of 99.13%, 98.35%, 93.26%, and 99.81%, respectively, for accuracy, precision, recall, and specificity. Nonetheless, M-14 attains a remarkable 98.42% accuracy rate, albeit with a lower recall value of 87.52%, indicating its potential limitations in identifying affirmative situations. M-12 and M-13 are clearly better models overall.

4.4 Major Findings and Conclusion

In this chapter, we compared and analyzed various machine learning, deep learning, and hybrid algorithms. These models are examined by using various preprocessing and traditional feature selection methods. According to this work, performance of classic machine learning models like RF, SVM, and DT were surpassed by deep learning models like CNN, CRNN, and GRU. Additionally, it has been shown that hybrid models—RF+ CNN and RF+LSTM have exceptional accuracy due to their ability to combine the strengths of ML and DL. Further research can be focussed on creating hybrid models by employing optimization techniques to optimize the hyperparameters of models.

Chapter 5

Proposed Methodology

This chapter explains the suggested nature-inspired algorithm for feature selection and ensemble classification models for disease prediction. Common problems faced by researchers in previous literature include lack of robustness, handling data imbalance, and high data dimensionality. Therefore, the suggested approach aims to implement an enhanced approach to predict heart diseases. Different stages involved in our framework are:

- Inter-Quartile Range based outlier detection
- Min-Max Normalization based pre-processing
- Hybrid of Genetic Algorithm and Cuckoo Search Algorithm based feature selection
- Ensemble Classification based Prediction
- Genetic Algorithm-based hyperparameter tuning.

Figure 5.1 visually represents the overall framework of the proposed approach.

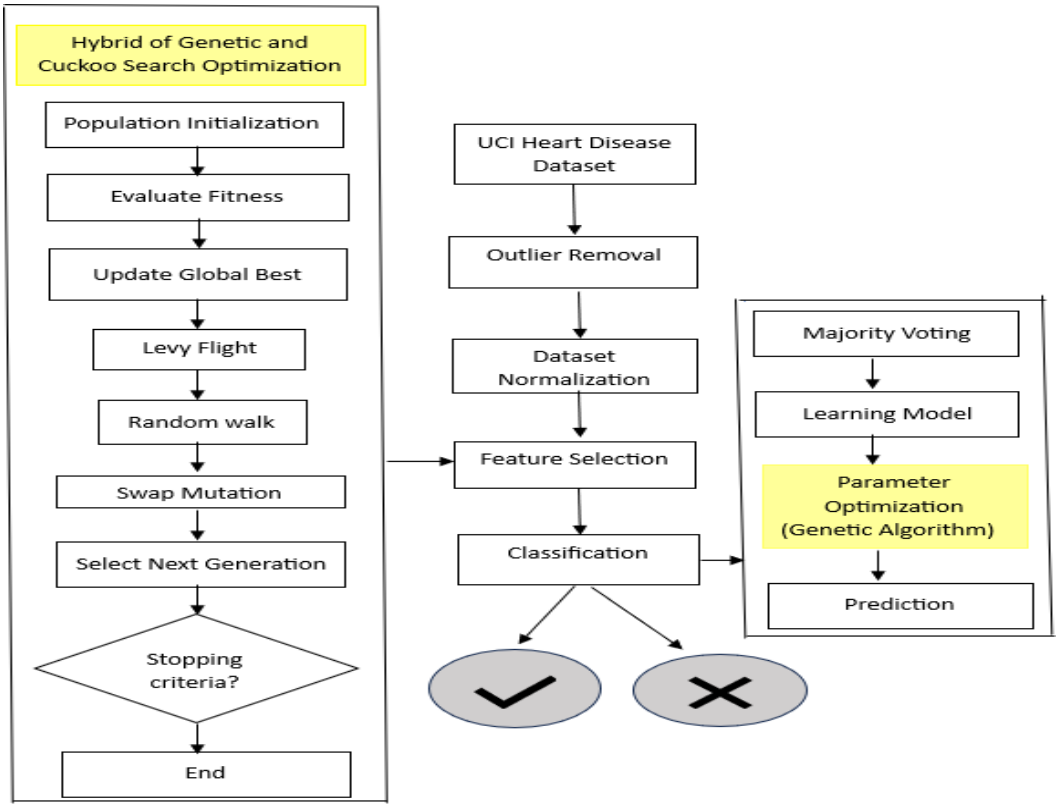


Figure 5.1 Flowchart of proposed model

5.1 Dataset Description

In this work, the UCI heart disease dataset has been used. This dataset has 14 attributes and 303 instances [3]. Table 5.1 shows the attributes and the corresponding values of the data.

Table 5.1 Dataset values

Attribute Name	Value Range
Age	29-79
Sex	0,1
CP	1-4
Trestbps	94-200
Chol	126-564
Fbs	0,1
Restecg	0-2
Thalach	71-202
Exang	0,1
OldPeak	1-3
Slope	1-3
Ca	0-3
Thal	3,6,7
Num	0,1

To summarize this dataset, we can conclude that patients aged 29-79 were included, where females were encoded as 0, while males as 1. There are four types of chest pain: Typical Angina (Type 1), Atypical Angina (Type 2); non-Anginal pain (Type 3); and Asymptomatic (Type 4). 'Trestbps' represents Blood pressure and 'Chol' represents Cholesterol level. The feature 'Fbs' (Fasting blood sugar) has two values 0 or 1. 'Restecg' indicates electrocardiographic results. 'Thalach' represents the maximum achieved heart rate, while Exercise-induced Angina, 'Exang', has a value of 0 or 1 depending on pain. 'OldPeak' shows ST depression induced by exercise [4]. The 'Slope' feature described slope of the peak of the ST segment and has values 3,6 or 7. 'Thal' represents 3 types of defects namely normal, fixed and reversible. Target variable, 'Num', indicates if patient has heart disease (1) or not (0).

5.2 Preprocessing

Data preprocessing is crucial step in the data analysis process. It includes preparing the data to assure the consistency of the dataset. Preprocessing can significantly enhance the performance of machine learning models. This section describes the preprocessing steps used in this work, including outlier removal, normalization, and feature selection.

5.1.1 Outlier removal

Outliers are data instances that significantly deviate from the expected range of values, causing bias in analysis and leading to wrong conclusions. In this study, outliers are identified and eliminated using the Inter-Quartile Range (IQR). The IQR represents statistical dispersion and is given by the formula IQR = Q3 - Q1, where Q1 is the first quartile and Q3 is the third quartile. The upper bound was calculated as Q3 + 1.5*IQR, while the lower bound as Q1 - 1.5*IQR. Data points that lie outside of these bounds are considered outliers and are removed [12]. This step ensures that the remaining data represents underlying patterns more closely, thereby increasing the reliability of our analysis. Figure 5.2 illustrates outliers present in our data.

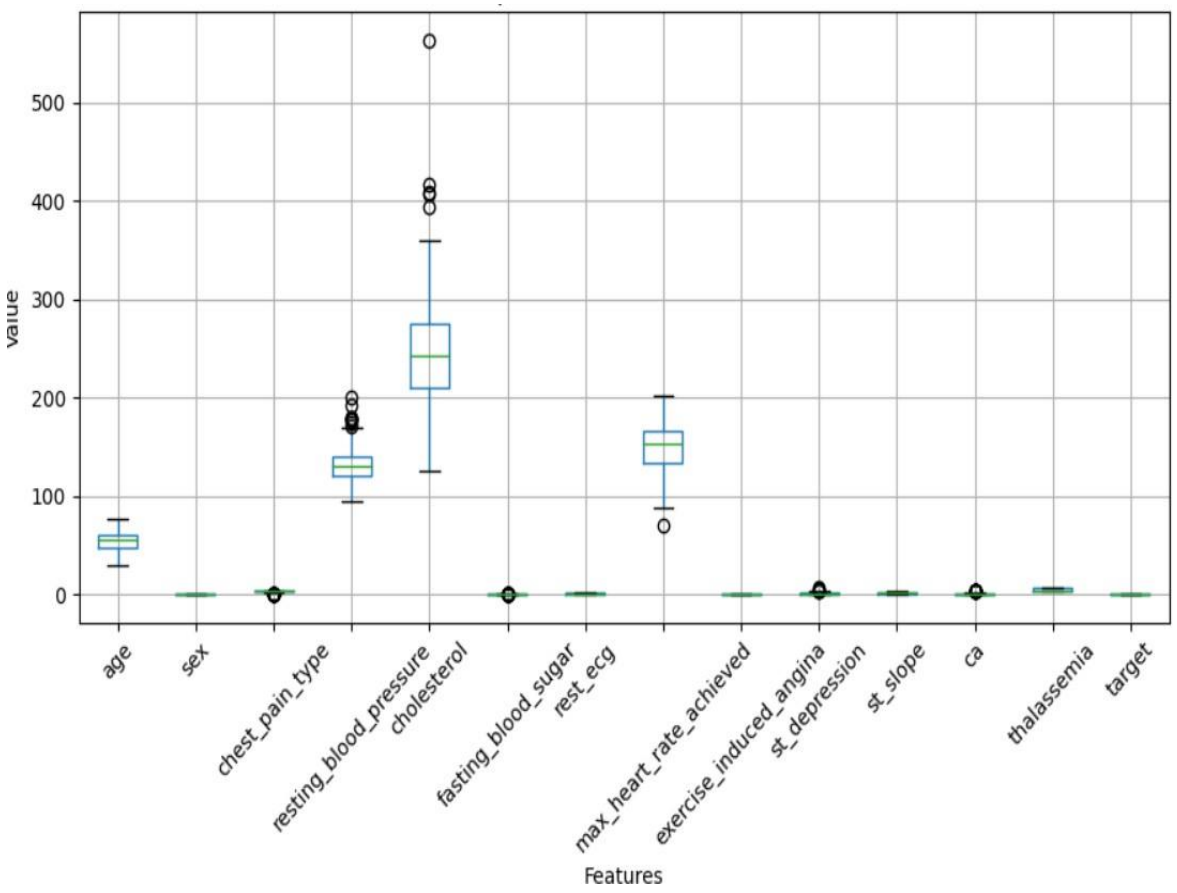


Figure 5.2 Box Plots showing Outliers

5.1.2 Normalization

After removing all the outliers, normalization is employed to standardize the data and prepare it for subsequent analysis. Normalizing the data ensures that the model doesn't get biased, which leads to enhanced accuracy. We applied two methods: Linear Interpolation Normalization and Min-Max Normalization. Among these, Min-Max Normalization yielded better results, so it was used for further research work.

Min-Max Normalization

Min-Max normalization transforms our data into a uniform scale. This technique balances the influence of all the attributes of data by preventing any single feature from overpowering others. Normalization is done using formula:

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}$$

5.1.3 Feature Selection

To identify the most optimal features for the target variable and enhance the performance of the model, a feature selection method is employed [8]. Three nature-inspired feature selection algorithms were utilized: Cuckoo Search Algorithm (CSA), Genetic Algorithm (GA), and Flower Pollination Algorithm (FPA). Among these, a hybrid approach combining GA and CSA yielded the best results, and it was subsequently used for further research.

Hybrid Approach (GA+CSA)

A hybrid strategy combining GA and CSA yielded the best results among the feature-selection algorithms used. This hybrid approach enhances feature selection by leveraging the strengths of both algorithms. CSA introduces diversity through its cuckoo-inspired search behaviour and Levy flight mechanism, while GA offers a strong mechanism for exploring the search space through natural selection [50]. By integrating these two algorithms, the hybrid approach effectively identifies the most relevant features, resulting in better model performance. The steps involved in the hybrid approach are shown in Algorithm 1.

Algorithm 1. Hybrid GA+CSO algorithm

1. Generate the initial population randomly
2. for t = 1, 2, ..., MaxGenerations do
3. Calculate fitness for each individual
4. for i = 1, 2, ..., PopSize do
5. Create solution $x_i(t+1)$ based on Lévy flight
6. $x_i(t+1) = x_i(t) + \alpha_i(t+1)$ * Levy(s, λ) * $(x_i(t) - x_{gbest})$ [50]
7. end for
8. Update global best x_{gbest}
9. for i = 1, 2, ..., PopSize do
10. Update $x_i(t+1)$ using random walk
11. if rand > threshold then
12. $x_i(t+1) = x_i(t)$ + rand * $(x_k(t) - x_j(t))$ [50]
13. end if
14. Repair the solution
15. Calculate fitness and update the solution
16. end for
17. Update global best x_{gbest}
18. if x_{gbest} does not improve for δ attempts then
19. Apply swap mutation
20. $x_i(t+1) = x_i(t) + \chi_\mu$ [50]
21. end if
22. Evaluate and update solutions
23. end for
24. Store the final solution x_{gbest}

Figure 5.3 gives flowchart of steps involved in the hybrid approach of GA+CSA. This hybrid feature selection approach identifies the best features by thoroughly exploring the search space. Table 5.2 shows parameters of algorithms used in hybrid feature selection method. This approach enhanced the predictivity of our model and was used for further research.

Table 5.2 Parametric study of GA+CSA hybrid as feature selection

Hyperparameters of GA+CSO	Value
Number of generations	100
Population size	50
Mutation Probability	0.05
Random walk probability	0.25
Crossover Probability	0.9
Delta (Number of generations to check)	5
Levy flight lambda	1.5

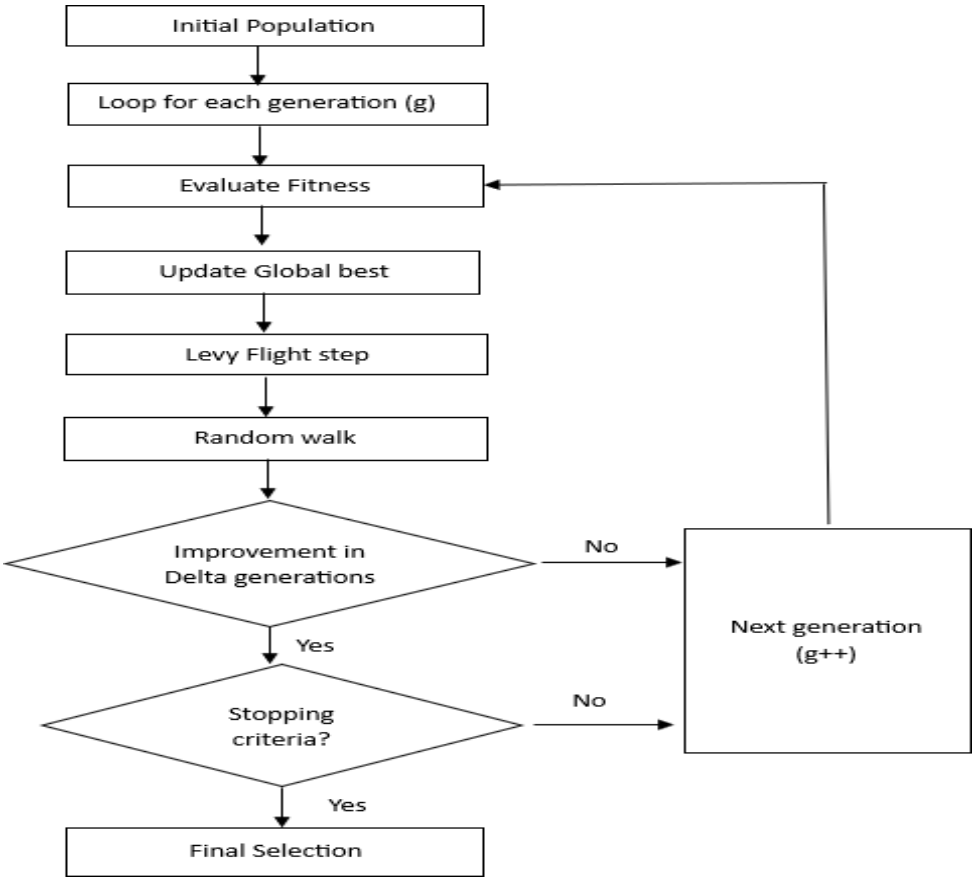


Figure 5.3 Flowchart of hybrid GA+CSA

5.3 RF+CNN Ensemble Based Classification

In this step, the classifier is trained with the features selected in the previous step to predict heart disease. We tried various machine learning models like, RF, LR, XGB, NB, deep learning models like CNN, RNN, GRU, Bi-LSTM, and ensemble approaches like RF+XGB+NB, XG+RF+CNN, and many more.

Out of these, RF+CNN+Majority voting gave the best results. This proposed model leverages the advantages of both RF and CNN to form a comprehensive predictive framework. The complete details of the proposed model are provided below.

Random Forest (RF)

Random forest is an ensemble learning approach that utilizes multiple decision trees and aggregates their outputs using majority voting to enhance the model's performance. A key concept used in Random Forest is Bootstrap aggregation (Bagging), where a decision tree is fed with bootstrapped data (randomly selected subsets with replacement). This technique reduces the chances of overfitting as it introduces randomness among individual trees. In our study, the RF model is configured with specific parameters optimized through the use of a Genetic Algorithm: a maximum depth of 43 and 194 estimators. These parameters ensure consistent performance by creating a balance between generalization and model complexity.

Convolutional Neural Network (CNN)

A CNN is a deep learning model used in image recognition, and object detection by extracting intricate patterns and spatial dependencies [33]. In this study, the CNN architecture begins with a 1D convolutional layer having 34 filters, each scanning input attribute to find local patterns. After this layer, Rectified Linear Unit activation is used to capture complex relationships within data. Subsequently, a max pooling layer with a pool size of 2 down samples the data. Two fully connected layers further enhance the architecture. There are 50 units with ReLU activation in the first dense layer and one unit with a sigmoid activation function for final binary prediction in the last dense output layer.

In the majority voting mechanism, predictions from both models are combined, and the final output is determined based on the majority decision. This methodology improves the overall accuracy and robustness by combining the strengths of both RF and CNN models.

5.4 Parameter tuning

To enhance the performance of our model, a Genetic Algorithm is used for hyperparameter tuning. Subsection 3.3 explains the steps involved in GA.

Table 5.3 depicts the parameters of the genetic algorithm used for hyperparameter tuning of the model while Table 5.4 shows the hyperparameters of the model as chosen by the GA. The final RF and CNN models were trained using these selected parameters and the final ensemble was created using majority voting procedure.

Table 5.3 Parameters of GA

Hyperparameters	**Value**
Population size	10
Crossover Probability	0.9
Number of generations	20
Mutation Probability	0.05

Table 5.4 Hyperparameters of models chosen by GA

Model	**Hyperparameters**	**Value**
RF	Number of estimators	190
	Maximum depth	43
CNN	Number of filters	34
	Kernal size	2

5.5 Results and Discussions

This section confirms effectiveness and outcomes of proposed hybrid feature selection-based ensemble methodology to detect heart diseases. A summary of the data used is provided in the succeeding sections. For our research, the dataset was divided into two subsets: a training set (80%) and a testing set (20%).

Figure 5.4 demonstrates the convergence comparison of the proposed approach and existing feature selection techniques. The convergence plot shows how effectively the optimizer provides optimal solutions with varying numbers of iterations. From the plot, it is evident that GA+CSO demonstrates superior performance, consistently showing faster convergence compared to other optimizers. Moreover, the stability and minimal fluctuations in GA+CSO make it the most reliable choice.

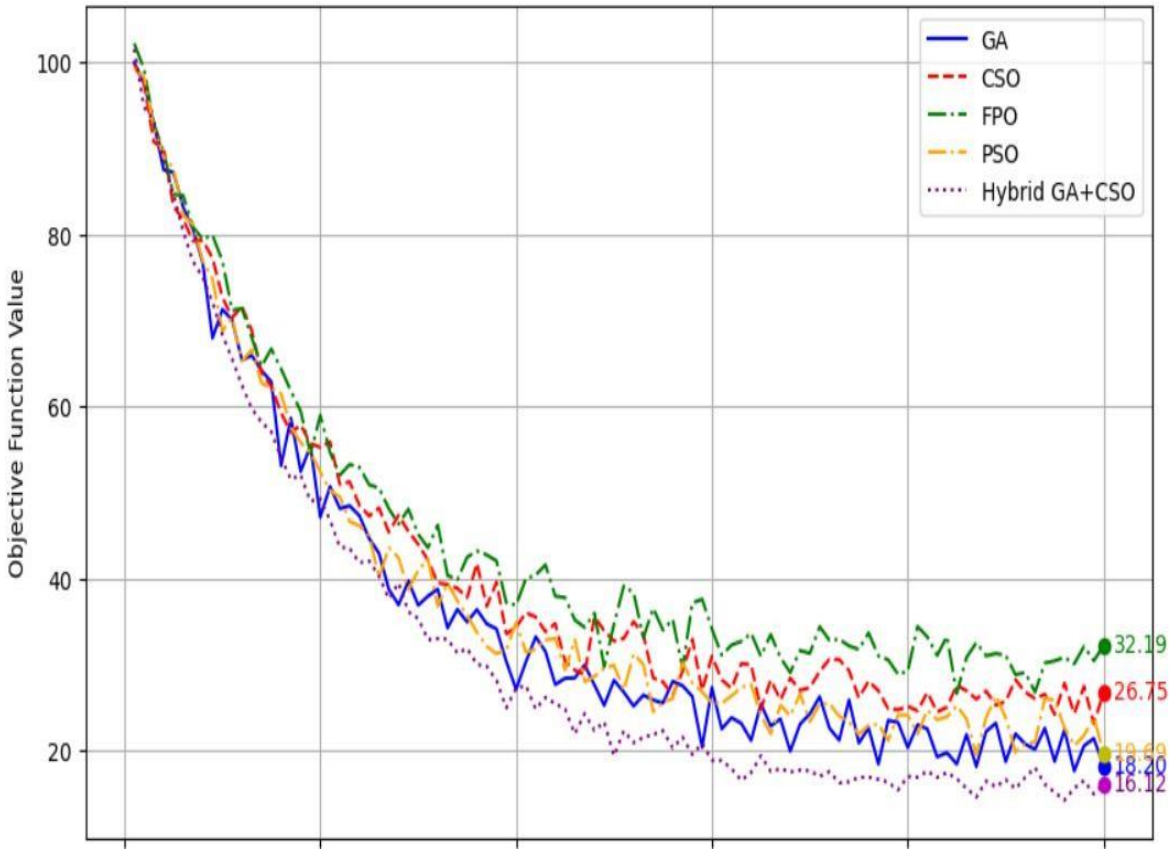


Figure 5.4 Convergence analysis

Figure 5.5 and Figure 5.6 compare the accuracy of the proposed model with the traditional linear machine learning and ensemble models with no feature selection and no parameter tuning. The corresponding values are given in Table 5.5 and Table 5.6 respectively.

Table 5.5 Comparison among linear ML models

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Specificity (%)	ROC AUC(%)
Logistic Regression	90	84.62	91.67	88.00	88.89	94.91
Linear SVC	90	87.50	87.50	87.50	91.67	94.56
Naïve Bayes	86.67	83.33	83.33	83.33	88.89	95.49
KNN	83.33	81.82	75	78.26	88.89	94.21
Proposed	**95**	**95.65**	**91.7**	**93.61**	**97.22**	**95.02**

Table 5.6 Comparison among ensemble models

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Specificity (%)	ROC AUC(%)
Random Forest	88.33	84.00	87.50	85.71	88.89	94.33
Gradient Boosting	80.00	71.43	83.33	76.92	77.78	89.93
AdaBoost Classifier	83.33	76.92	83.33	80.00	83.33	93.63
Decision Tree	78.33	70.37	79.17	74.51	77.78	78.47
Proposed	**95**	**95.65**	**91.7**	**93.61**	**97.22**	**95.02**

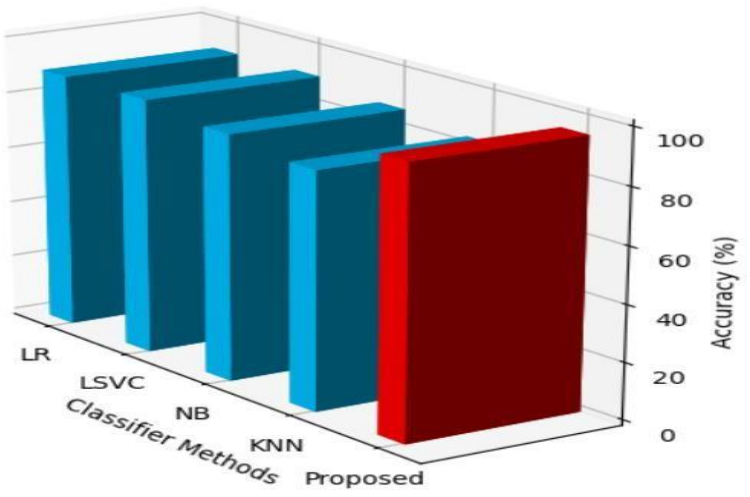


Figure 5.5 Accuracies for linear ML models

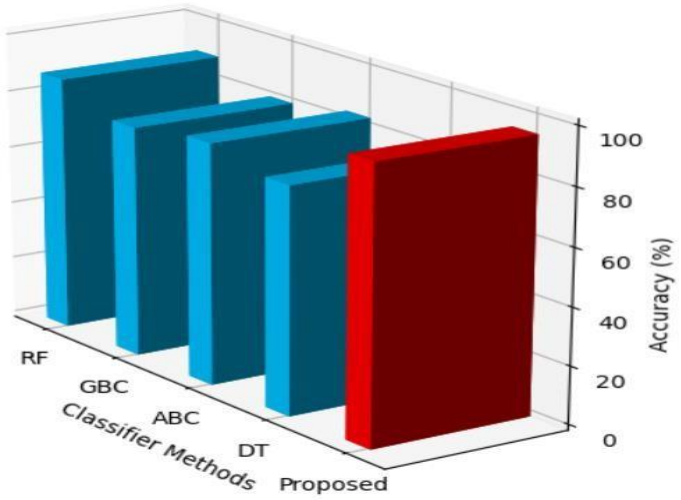


Figure 5.6 Accuracies for ensemble models

Based on Tables 5.5 and 5.6, the proposed model performs best compared to existing linear ML and ensemble models in terms of all performance metrics. Among linear models, following the proposed model, LR and linear SVC achieve same accuracy of 90%, while KNN is the least performing model. In case of ensemble models, RF is the best performing model whereas DT performs the least.

Using the hybrid GA+CSA as a feature selection method, Tables 5.7 compares the accuracies and recall of the proposed model with existing models, with parameter tuning performed by GA and CSA. The visual representations of these comparisons are shown in Figures 5.7 and 5.8

Table 5.7 Comparison between different tuning methods

	Genetic Algorithm		Cuckoo Search Algorithm	
MODELS	**Accuracy**	**Recall**	**Accuracy**	**Recall**
RF	84.38	87.50	93.75	90.62
LR	86	84	79.6	78.12
NB	77	69	79.5	75
SVM	88	88	79.6	75
RF+CNN (Proposed)	**95**	**91.7**	**89.06**	**81.25**

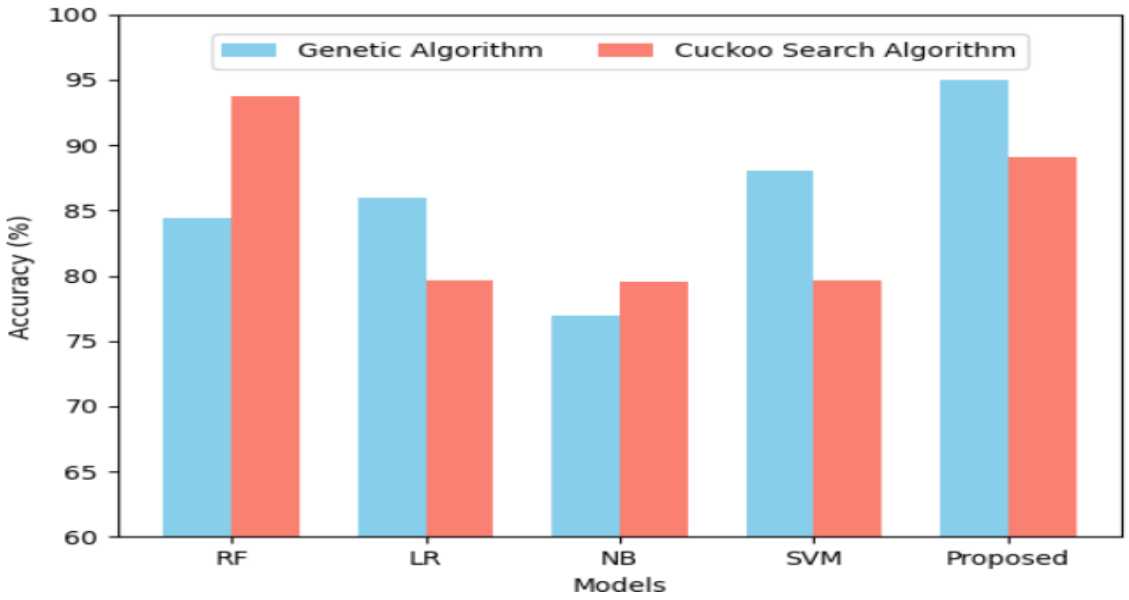


Figure 5.7 Comparison between accuracies for different tuning algorithms

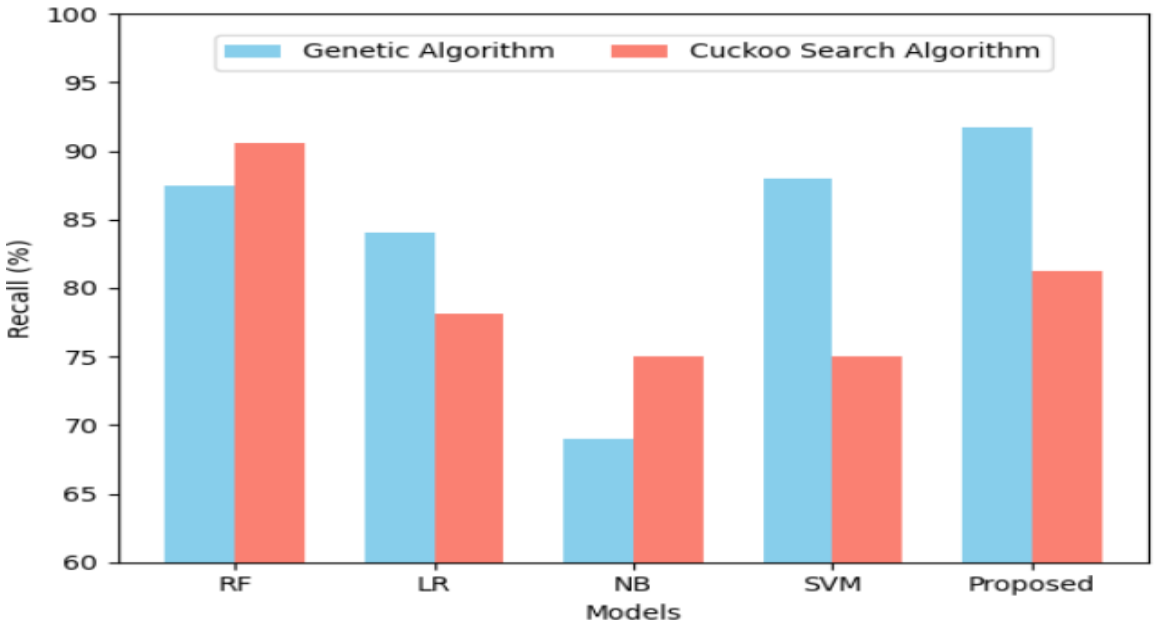


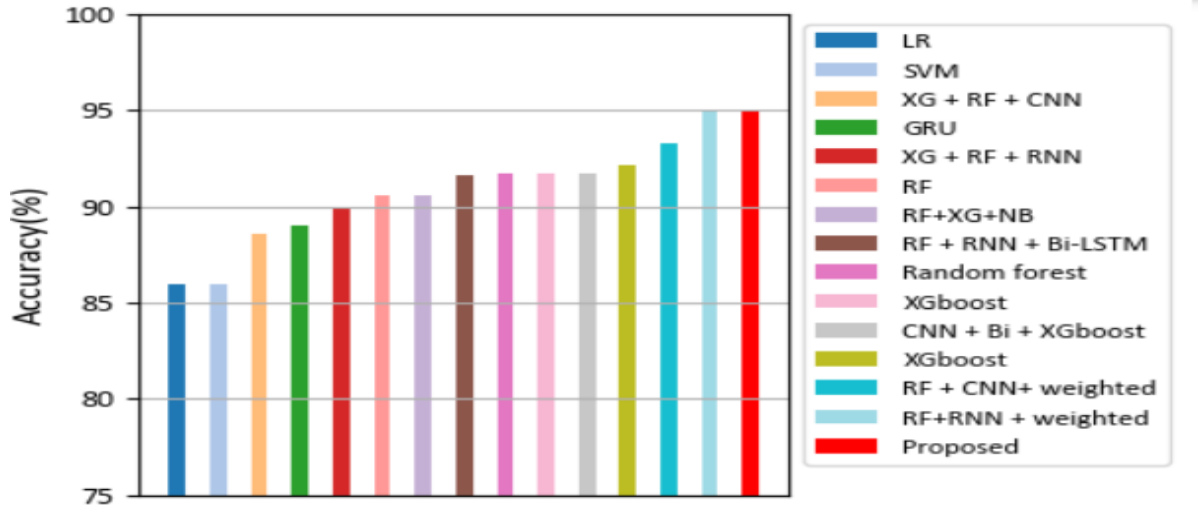
Figure 5.8 Comparison between recall for different tuning algorithms

From Figure 5.7 and Figure 5.8, it is evident that the proposed model achieves the highest accuracy (95%) and recall (91.7%) with GA tuning compared to other models. Following the proposed model, SVM is the second-best performing model with GA tuning, achieving both accuracy and recall of 88%. RF has the highest accuracy of 93.75% under GA, while NB performs poorly with 77% accuracy and 69% recall. Using CSA tuning, RF emerges as the best performer with 93.75% accuracy and 90.62% recall, followed by proposed RF+CNN with majority voting having an accuracy of 89.06% and recall of 81.25%. LR, NB, and SVM exhibit lower performance metrics with slight variations in their results.

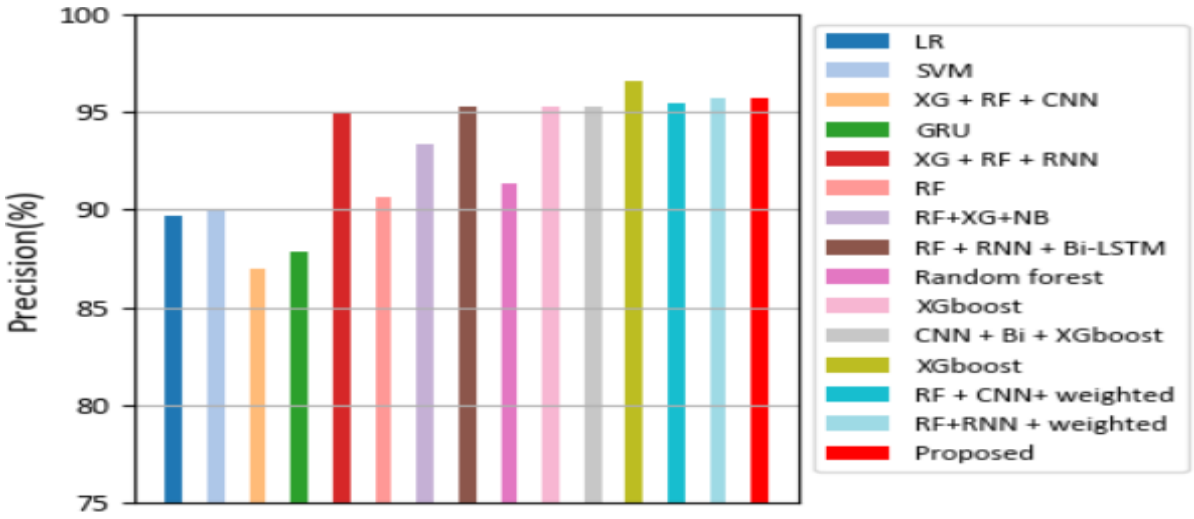
Table 5.8 and Figure 5.9 provide a comprehensive comparison of the proposed model with various existing ML models with different feature selection and parameter tuning methodologies.

Table 5.8 Overall summary of models with different feature selection and parameter tuning methods

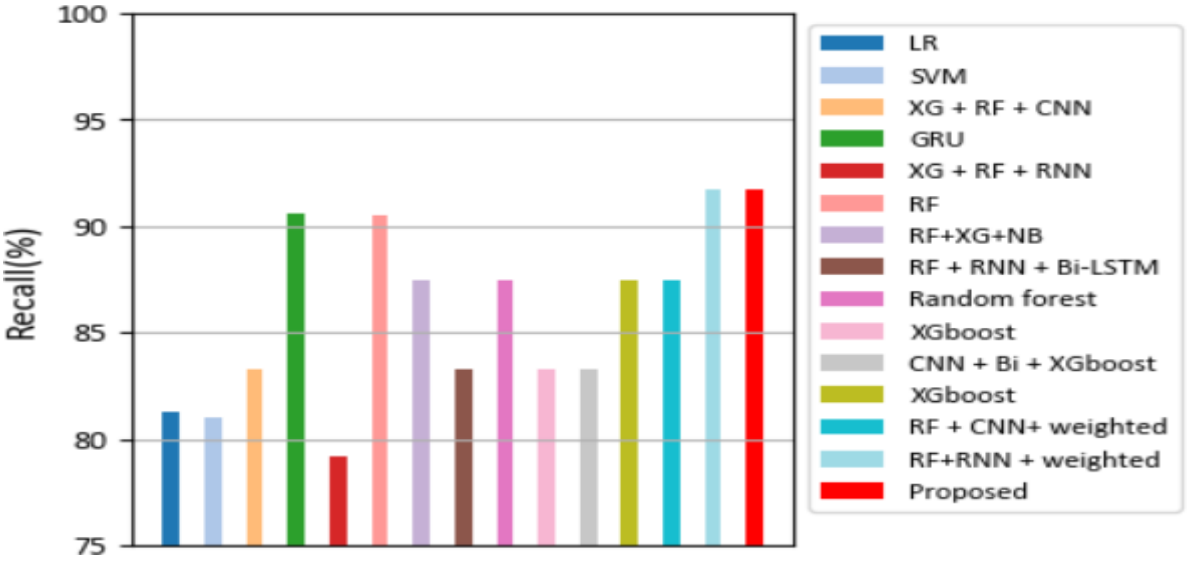
Models	**Feature Selection**	**Tuning**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Specificity**	**ROC**
LR	CSO	CSO	85.93	89.65	81.3	85.24	90.62	96.3
SVM	CSO	GA	86	90	81	85	91	92
XG + RF + CNN	GA+CSO	GA	88.6	86.96	83.3	85.11	91.67	94.49
GRU	CSO	GA	89	87.87	90.6	89.23	87.5	97.11
XG + RF + RNN	GA+CSO	GA	90	95	79.2	86.36	97.22	96.41
RF	CSO	CSO	90.6	90.6	90.5	90.6	90.6	97.11
RF+XG+NB	CSO	CSO	90.6	93.3	87.5	93.75	90.32	94.2
RF+ RNN+ BiLSTM	GA+CSO	GA	91.66	95.23	83.3	88.88	97.22	94.68
Random forest	GA+CSO	GA	91.67	91.3	87.5	89.36	94.44	96.53
XGBoost	GA+CSO	GA	91.67	95.24	83.3	88.89	97.22	94.91
CNN+BiLSTM+XGB	GA+CSO	GA	91.67	95.24	83.3	88.89	97.22	95.60
XGBoost	CSO	CSO	92.18	96.55	87.5	91.8	96.87	96.3
RF+CNN + weighted	GA+CSO	GA	93.3	95.45	87.5	91.3	97.22	94.10
RF +RNN+ weighted	GA+CSO	GA	95	95.65	91.7	93.61	97.22	94.06
Proposed	GA+CSO	**GA**	**95**	**95.65**	**91.7**	**93.61**	**97.22**	**95.02**


100
95
90
85
80
75
Accuracy(%)
LR
SVM
XG + RF + CNN
GRU
XG + RF + RNN
RF
RF+XG+NB
RF + RNN + Bi-LSTM
Random forest
XGboost
CNN + Bi + XGboost
XGboost
RF + CNN+ weighted
RF+RNN + weighted
Proposed

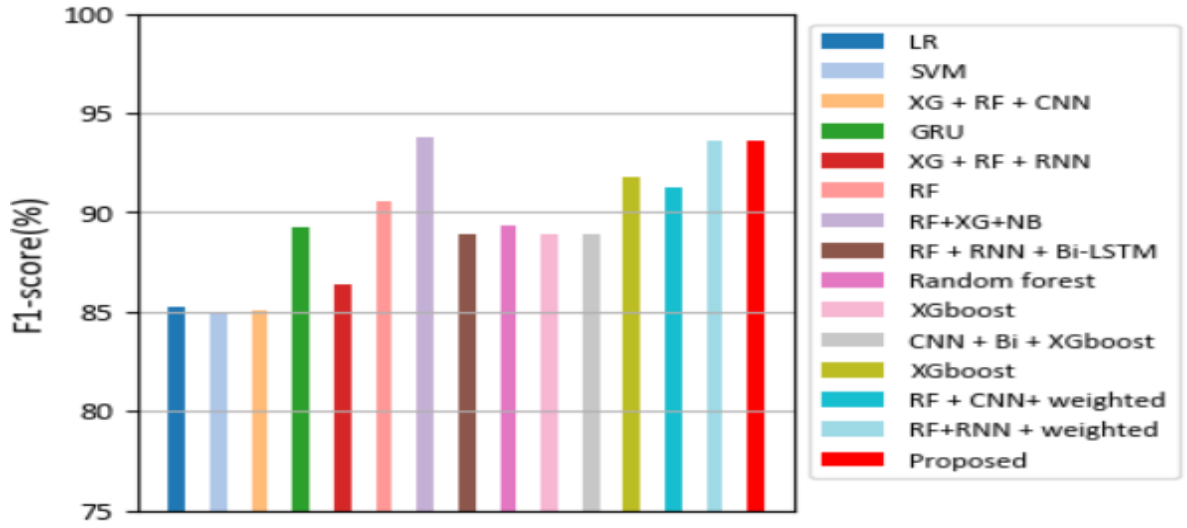
(a)


100
95
90
85
80
75
Precision(%)
LR
SVM
XG + RF + CNN
GRU
XG + RF + RNN
RF
RF+XG+NB
RF + RNN + Bi-LSTM
Random forest
XGboost
CNN + Bi + XGboost
XGboost
RF + CNN+ weighted
RF+RNN + weighted
Proposed

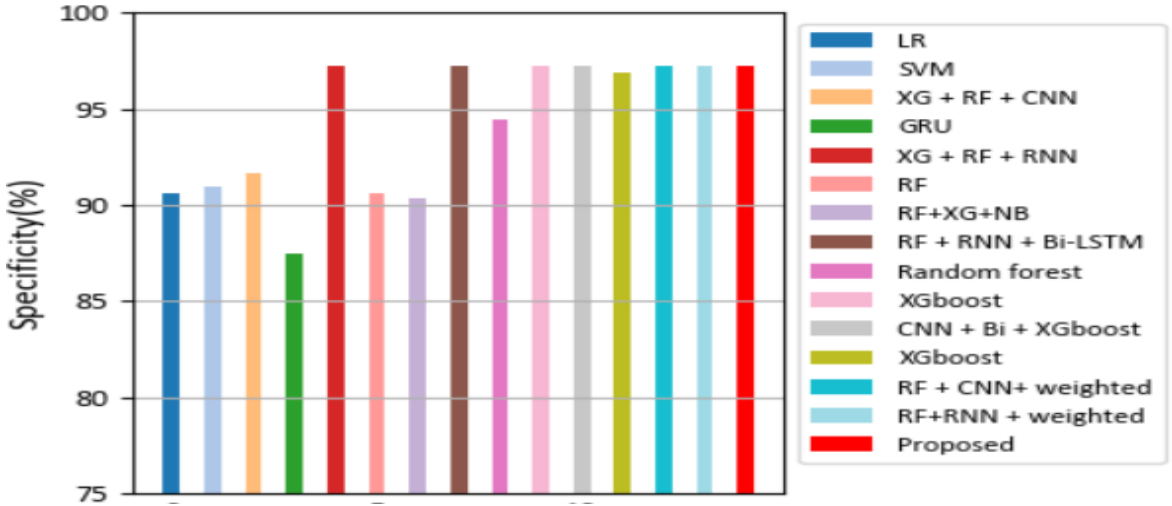
(b)


100
95
90
85
80
75
Recall(%)
LR
SVM
XG + RF + CNN
GRU
XG + RF + RNN
RF
RF+XG+NB
RF + RNN + Bi-LSTM
Random forest
XGboost
CNN + Bi + XGboost
XGboost
RF + CNN+ weighted
RF+RNN + weighted
Proposed

(c)



(d)



(e)

Figure 5.9 Overall Performance analysis

From Figure 5.9(a), it can be seen that in terms of accuracy, the best-performing models are the Proposed model and the RF+ RNN + weighted ensemble, both with an accuracy of 95%. These are followed by the RF + CNN + weighted ensemble model, with an accuracy of 93.3%, and the XG Boost model, tuned with CSO, with an accuracy of 92.18%. In terms of precision, Figure 5.9(b) validates that the top-performing model is the XG Boost model, tuned with CSO, having a precision of 96.55%. This is followed by the Proposed model and the RF + RNN + weighted ensemble, both with a score of 95.65%. The RF + CNN + Majority voting, RF + RNN + weighted ensemble, and RF + CNN + weighted ensemble models each have slight variations in their precision values. In terms of recall, as shown in Figure 5.9(c), the Proposed model and the RF + CNN + weighted ensemble have the highest value of 91.7%, followed by the GRU and RF models, both tuned with CSO, with scores of 90.6% and 90.5%, respectively. Figure 5.9(d) demonstrates that in terms of specificity, the highest value of 97.22% is achieved by the Proposed model, along with some other models.

Therefore, the proposed RF + CNN model with majority voting demonstrates superior performance across various metrics compared to traditional machine learning models and ensemble methods. The model achieved an accuracy of 95%, which is among the highest in the comparison table. It has a precision of 95.65%, indicating a high true positive rate. The recall score of 91.7% outperforms all other models, suggesting that our proposed model effectively identifies many positive cases. A higher recall value is crucial in disease prediction as it ensures that the maximum number of disease cases are detected correctly. The F1 score is 93.61%, reflecting a good balance between recall and precision. With a specificity score of 97.22%, the proposed approach is highly effective at correctly identifying negative cases. Additionally, the ROC score of 95.02% confirms a good trade-off between the false positive and true positive rates.

Overall, the proposed combination yields a model that is both precise and robust across various evaluation metrics.

5.6 Major Findings and Conclusion

This chapter describes the proposed method using a hybrid approach that combines the Cuckoo Seach algorithm and Genetic Algorithm for feature selection, followed by an ensemble of Convolutional Neural Networks and Random Forest using majority voting for prediction. The dataset used in this research is UCI Heart Disease Dataset. The pre-processing techniques are removing outliers using the Interquartile range (IQR) and standardization using Min-max normalization. The hybrid GA+CSA was used for selecting the most relevant features and hyperparameters were optimized using GA. Using the proposed approach, 95% accuracy, 95.65% precision, 91.7% recall, 93.61% F1-score, 97.22% specificity, and 95.02% ROC AUC has been achieved.

Chapter 6

Overall Conclusion and Future Work

This thesis has examined the application of machine learning (ML), deep learning (DL), and hybrid models for predicting heart disease on MIT-BIH Arrhythmia dataset. Through rigorous comparative analysis, deep learning algorithms like CNN, CRNN, and GRU demonstrated higher performance over traditional ML algorithms such as RF, SVM, and DT. Hybrid models showcased effective combination of ML and DL techniques, highlighting their potential for precise diagnosis.

The proposed hybrid approach, combining the Cuckoo Search Algorithm (CSA) and Genetic Algorithm (GA) for feature selection, followed by an ensemble of CNN and Random Forest using majority voting, achieved 95% accuracy and proved out to be the robust model.

Future work will explore optimization methods to enhance model's performance in predicting heart disease. Moreover, we plan to validate our approach using other datasets and include more sophisticated deep-learning models to enhance the robustness of the proposed method.

List of Publications

1. Gupta, I., Bajaj, A. and Sharma, V., 2024. Comparative analysis of machine learning algorithms for heart disease prediction. *International Journal of Hybrid Intelligent Systems*, pp.1-15.
2. Gupta, I., Bajaj, A. and Sharma, V., 2024. A Survey of Machine Learning Algorithms for Heart Disease Prediction. *14th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2023)* - Presented
3. Gupta, I., Bajaj, A. and Sharma, V., 2024. Heart Disease Prediction Using a Hybrid Feature Selection and Ensemble Learning Approach. *Soft Computing (Springer)* - Communicated

References

1. Yang, J., & Guan, J. (2022). A heart disease prediction model based on feature optimization and smote-Xgboost algorithm. *Information*, *13*(10), 475.
2. Singh, P., & Singh, D. P. (2023, March). A Novel Framework for Prediction of Heart Disease Using Hybrid Classifier in the Cloud Environment. In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 702-707). IEEE.
3. Dileep, P., Rao, K. N., Bodapati, P., Gokuruboyina, S., Peddi, R., Grover, A., & Sheetal, A. (2023). An automatic heart disease prediction using cluster-based bi-directional LSTM (C-BiLSTM) algorithm. *Neural Computing and Applications*, *35*(10), 7253-7266.
4. Subathra, R., & Sumathy, V. (2024). An offbeat bolstered swarm integrated ensemble learning (BSEL) model for heart disease diagnosis and classification. *Applied Soft Computing*, *154*, 111273.
5. Atehortúa, A., Gkontra, P., Camacho, M., Diaz, O., Bulgheroni, M., Simonetti, V., ... & Lekadir, K. (2023). Cardiometabolic risk estimation using exposome data and machine learning. *International Journal of Medical Informatics*, *179*, 105209.
6. Zhang, L., Niu, M., Zhang, H., Wang, Y., Zhang, H., Mao, Z., ... & Wang, C. (2022). Nonlaboratory-based risk assessment model for coronary heart disease screening: Model development and validation. *International Journal of Medical Informatics*, *162*, 104746.
7. Itoo, N. N., & Garg, V. K. (2022, March). Heart disease prediction using a stacked ensemble of supervised machine learning classifiers. In *2022 International Mobile and Embedded Technology Conference (MECON)* (pp. 599-604). IEEE.
8. Kapila, R., Ragunathan, T., Saleti, S., Lakshmi, T. J., & Ahmad, M. W. (2023). Heart disease prediction using novel Quine McCluskey binary classifier (QMBC). *IEEE Access*, *11*, 64324–64347.
9. Sulthana, R., Jaithunbi, A. K., & Sunraja, P. (2023). Application of machine learning algorithms in predicting heart disease in patients. In *2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies* (pp. 1-4). IEEE.
10. Gaikwad, M. J., Asole, P. S., & Bitla, L. S. (2022). Effective study of machine learning algorithms for heart disease prediction. In *2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control* (pp. 1-6). IEEE.
11. Jadhav, S. R., Kulkarni, R., Yendralwar, A., Pujari, P., & Patwari, S. (2023). Monitoring and predicting heart diseases using machine learning techniques. In *2023 IEEE 8th International Conference for Convergence in Technology* (pp. 1-4). IEEE.
12. Yang, H., Chen, Z., Yang, H., & Tian, M. (2023). Predicting coronary heart disease using an improved LightGBM model: Performance analysis and comparison. *IEEE Access*, *11*, 23366-23380.
13. Vijaya, J. (2023). Heart disease prediction using clustered genetic optimization algorithm. In *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics* (pp. 1072-1077). IEEE.

14. Islam, M. T., Rafa, S. R., & Kibria, M. G. (2020). Early prediction of heart disease using PCA and hybrid genetic algorithm with k-means. In *2020 23rd International Conference on Computer and Information Technology* (pp. 1-6). IEEE.
15. Abdellatif, A., Abdellatef, H., Kanesan, J., Chow, C. O., Chuah, J. H., & Gheni, H. M. (2022). An effective heart disease detection and severity level classification model using machine learning and hyperparameter optimization methods. *IEEE Access, 10*, 79974-79985.
16. Sharma, S., & Parmar, M. (2020). Heart diseases prediction using deep learning neural network model. *International Journal of Innovative Technology and Exploring Engineering, 9*(3), 2244-2248.
17. Chamundeshwari, Biradar, N., & Udaykumar. (2023). Adaptive despeckling and heart disease diagnosis by echocardiogram using optimized deep learning model. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 11*(1), 1-17.
18. Jain, A., Rao, A. C. S., Jain, P. K., & Hu, Y. C. (2023). Optimized Levy flight model for heart disease prediction using CNN framework in big data application. *Expert Systems with Applications, 223*, 119859.
19. Hussain, S., Nanda, S. K., Barigidad, S., Akhtar, S., Suaib, M., & Ray, N. K. (2021). Novel deep learning architecture for predicting heart disease using CNN. In *2021 19th OITS International Conference on Information Technology (OCIT)* (pp. 353-357). IEEE.
20. Verma, K., Bartwal, A. S., & Thapliyal, M. P. (2021). A genetic algorithm-based hybrid deep learning approach for heart disease prediction. *Journal of Mountain Research, 16*(3), 179-187.
21. Sugendran, G., & Sujatha, S. (2023). Earlier identification of heart disease using enhanced genetic algorithm and fuzzy weight based support vector machine algorithm. *Measurement: Sensors, 28*, 100814.
22. Zhao, N., Li, X., Ma, Y., Wang, H., Lee, S. J., & Wang, J. (2024). Improved stacked ensemble with genetic algorithm for automatic ECG diagnosis of children living in high-altitude areas. *Biomedical Signal Processing and Control, 87*, 105506.
23. Nandakumar, P., & Narayan, S. (2022). Cardiac disease detection using cuckoo search enabled deep belief network. *Intelligent Systems with Applications, 16*, 200131.
24. Wang, Y., Liu, Q., Yang, Y., Wang, L., Song, X., & Zhao, X. (2023). Prognostic staging of esophageal cancer based on prognosis index and cuckoo search algorithm-support vector machine. *Biomedical Signal Processing and Control, 79*, 104207.

25. Malakar, S., Sen, S., Romanov, S., Kaplun, D., & Sarkar, R. (2023). Role of transfer functions in PSO to select diagnostic attributes for chronic disease prediction: An experimental study. *Journal of King Saud University-Computer and Information Sciences, 35*(9), 101757.
26. Liu, Z., Kou, J., Yan, Z., Wang, P., Liu, C., Sun, C., Shao, A., & Klein, B. (2024). Enhancing XRF sensor-based sorting of porphyritic copper ore using particle swarm optimization-support vector machine (PSO-SVM) algorithm. *International Journal of Mining Science and Technology*.
27. Wang, S., Ren, J., & Bai, R. (2023). A semi-supervised adaptive discriminative discretization method improving discrimination power of regularized naive Bayes. *Expert Systems with Applications, 225*, 120094.
28. Zhang, W., Di, X., Wei, G., Geng, S., Fu, Z., & Hong, S. (2024). Cardiac arrhythmia classification with rejection of ECG recordings based on uncertainty estimation from deep neural networks. *Neural Computing and Applications, 36*(8), 4047–4058.
29. Liu, Y., Liu, J., Tian, Y., Jin, Y., Li, Z., Zhao, L., & Liu, C. (2024). Pruned lightweight neural networks for arrhythmia classification with clinical 12-lead ECGs. *Applied Soft Computing*, 111340.
30. Yao, Q., Zhang, L., Zheng, W., Zhou, Y., & Xiao, Y. (2023). Multi-scale SE-residual network with transformer encoder for myocardial infarction classification. *Applied Soft Computing, 149*, 110919.
31. Karri, M., & Annavarapu, C. S. R. (2023). A real-time embedded system to detect QRS-complex and arrhythmia classification using LSTM through hybridized features. *Expert Systems with Applications, 214*, 119221.
32. Tiwari, S., Jain, A., Sapra, V., Koundal, D., Alenezi, F., Polat, K., Alhudhaif, A., & Nour, M. (2023). A smart decision support system to diagnose arrhythmia using ensembled ConvNet and ConvNet-LSTM models. *Expert Systems with Applications, 213*, 118933.
33. Korürek, M., & Nizam, A. (2008). A new arrhythmia clustering technique based on ant colony optimization. *Journal of Biomedical Informatics, 41*(6), 874–881.
34. Zhang, Z., Dong, J., Luo, X., Choi, K. S., & Wu, X. (2014). Heartbeat classification using disease-specific feature selection. *Computers in Biology and Medicine, 46*, 79–89.
35. Khalil, M., & Adib, A. (2020). An end-to-end multi-level wavelet convolutional neural networks for heart diseases diagnosis. *Neurocomputing, 417*, 187–201.
36. Bertsimas, D., Mingardi, L., & Stellato, B. (2021). Machine learning for real-time heart disease prediction. *IEEE Journal of Biomedical and Health Informatics, 25*(9), 3627–3637.
37. Chen, L., Lian, C., Zeng, Z., Xu, B., & Su, Y. (2023). Cross-modal multiscale multi-instance learning for long-term ECG classification. *Information Sciences,*119230.

38. Reddy, S. S. (2017). Optimal reactive power scheduling using cuckoo search algorithm. *International Journal of Electrical and Computer Engineering, 7(5),* 2349-2356.
39. Özbay, Y. (2009). A new approach to detection of ECG arrhythmias: Complex discrete wavelet transform-based complex valued artificial neural network. *Journal of Medical Systems, 33*(6), 435-445.
40. Sowmya, S., & Jose, D. (2022). Contemplate on ECG signals and classification of arrhythmia signals using CNN-LSTM deep learning model. *Measurement: Sensors, 24*, 100558.
41. Alamatsaz, N., Tabatabaei, L., Yazdchi, M., Payan, H., Alamatsaz, N., & Nasimi, F. (2024). A lightweight hybrid CNN-LSTM explainable model for ECG-based arrhythmia detection. *Biomedical Signal Processing and Control, 90*, 105884.
42. Begum, S. G., Priyadarshi, E., Pratap, S., Kulshrestha, S., & Singh, V. (2023). Automated detection of abnormalities in ECG signals using deep neural network. *Biomedical Engineering Advances, 5*, 100066.
43. Rahul, J., & Sharma, L. D. (2022). Automatic cardiac arrhythmia classification based on hybrid 1-D CNN and Bi-LSTM model. *Biocybernetics and Biomedical Engineering, 42*(1), 312-324.
44. Zhou, F. Y., Sun, Y. H., & Wang, Y. W. (2024). Inter-patient ECG arrhythmia heartbeat classification network based on multiscale convolution and FCBA. *Biomedical Signal Processing and Control, 90*, 105789.
45. Isin, A., & Ozdalili, S. (2017). Cardiac arrhythmia detection using deep learning. *Procedia Computer Science, 120*, 268-275.
46. Venkatesh, C., Prasad, B. V. V. S., Khan, M., Babu, J. C., & Dasu, M. V. (2024). An automatic diagnostic model for the detection and classification of cardiovascular diseases based on swarm intelligence technique. *Heliyon.*
47. Berkaya, S. K., Uysal, A. K., Gunal, E. S., Ergin, S., Gunal, S., & Gulmezoglu, M. B. (2018). A survey on ECG analysis. *Biomedical Signal Processing and Control, 43*, 216-235.
48. Naaz, A., & Singh, M. S. (2014). Feature extraction and analysis of ECG signal for cardiac abnormalities: A review. *International Journal of Engineering Research & Technology, 3*(11), 23-30.
49. Guo, S., Guo, D., Chen, L., & Jiang, Q. (2017). A L1-regularized feature selection method for local dimension reduction on microarray data. *Computational Biology and Chemistry, 67*, 92-101.
50. Bajaj, A., & Sangwan, O. P. (2021). Discrete cuckoo search algorithms for test case prioritization. *Applied Soft Computing, 110*, 107584.
51. Alzakari, S. A., Menaem, A. A., Omer, N., Abozeid, A., Hussein, L. F., Abass, I. A. M., Rami, A., & Elhadad, A. (2024). Enhanced heart disease prediction in remote healthcare monitoring using IoT-enabled cloud-based XGBoost and Bi-LSTM. *Alexandria Engineering Journal, 105*, 280-291.

52. Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, *11*(7), 1387.
53. Senthilkumar, G., Ramakrishnan, J., Frnda, J., Ramachandran, M., Gupta, D., Tiwari, P., Shorfuzzaman, M., & Mohammed, M. A. (2021). Incorporating artificial fish swarm in ensemble classification framework for recurrence prediction of cervical cancer. *IEEE Access*, *9*, 83876-83886.
54. Reddy, S. S. (2017). Optimal reactive power scheduling using cuckoo search algorithm. *International Journal of Electrical and Computer Engineering*, *7*(5), 2349-2356.