A

Dissertation Report

ON

DETECTION OF ARRHYTHMIA IN COVID-19 PATIENTS USING SUPERVISED LEARNING METHODS

Submitted in partial fulfillment of the requirement of University of Mumbai
For the Degree of

Master of Engineering
in
Information Technology

by

Mr. Yashodhan Ketkar

Under the Guidance of

Dr. Sushopti Gawade

DEPARTMENT OF INFORMATION TECHNOLOGY (PG)

**PILLAI COLLEGE OF ENGINEERING
NEW PANVEL - 410206**

UNIVERSITY OF MUMBAI

Academic Year 2020-21

DISSERTATION APPROVAL CERTIFICATE

This is to certify that the dissertation work entitled **"Detection of Arrhythmia in COVID-19 Patients Using Supervised Learning Methods"**, for **M.E. (Information Technology)** submitted to University of Mumbai by Yashodhan Ketkar, a bonafide student of Pillai College of Engineering, New Panvel has been approved for the award of Master of Engineering Degree in Information Technology.

Examiners:

Internal Examiner   **External Examiner**

(Signature)   (Signature)

Name: Name:

Date: Date:

DEPARTMENT OF INFORMATION TECHNOLOGY (PG)

PILLAI COLLEGE OF ENGINEERING
NEW PANVEL - 410206
UNIVERSITY OF MUMBAI

Academic Year 2020-21

DEPARTMENT OF INFORMATION TECHNOLOGY (PG)
PILLAI COLLEGE OF ENGINEERING
NEW PANVEL - 410206
UNIVERSITY OF MUMBAI
Academic Year 2020-21

CERTIFICATE

This is to certify that **Mr. Yashodhan Prakash Ketkar** has satisfactorily carried out the dissertation work entitled **"Detection of Arrhythmia in COVID-19 Patients Using Supervised Learning Methods"** for the degree of **Master of Engineering** in **Information Technology** of **University of Mumbai**

Dr. Sushopti Gawade   **Dr. Satishkumar Verma**
Project Guide   Head of Department
Computer Engineering   Information Technology
Pillai College of Engineering   Pillai College of Engineering

Dr. Sandeep M. Joshi
PRINCIPAL
Pillai College of Engineering, New Panvel

SYNOPSIS OF PROJECT WORK

Name of the Dissertation: Detection of Arrhythmia in Covid-19 Patients Using
                             Supervised Learning Methods

Student's Name: Mr. Yashodhan Prakash Ketkar

Class: M.E. (Information Technology)

College: Pillai College of Engineering, New Panvel

Semester: IV

University Registration Number: 2019016402240447

Date of Registration: 26-07-2019

Exam Fee Receipt No.:   CF 2019-2020/2021

                             CF2019-2020/2022

Name of the Guide: Dr. Sushopti Gawade

Semester	Exam Seat Number	Result
1st	4021718	7.81
2nd	6041462	9.52

Student	**Project Guide**
(Mr. Yashodhan Ketkar)	(Dr. Sushopti Gawade)

Table of Contents

List of Figures

List of Tables

List of Equations

Abstract

COVID-19 disease has become a pandemic over the past year. This disease is highly contagious and spread throughout many countries. This disease with combination of comorbidities can lead to serious consequences for its victims. Cardiovascular comorbidities are leading to serious illness and death in COVID-19 infection. These cardiovascular comorbidities can also manifest after COVID-19 infection. These Cardiovascular problems can be detected using supervised learning methods, faster and more efficiently than traditional methods. Method used in this report uses supervised learning methods for detection of these problems. The method automatically trains and selects the best model for data provided and stores this data for future predictions. This will allow medical staff to focus on patient care and develop solutions faster. The implemented system uses K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest Algorithm (RF), Multi-layer Perceptron (MLP) and Support Vector Machine (SVM). The proposed system measures performance of this algorithm with respect to accuracy, F1, recall, precision, ROC and prediction time for each item.

Chapter 1

Introduction

Chapter 1

Introduction

In the past few years, COVID-19 (Coronavirus Disease 2019) has rapidly become a pandemic. This disease, caused by a strain of the Corona virus, was discovered near the end of 2019, hence it was named COVID-19 disease. This disease primarily attacks the human respiratory system. The infected respiratory system leads to various complicated respiratory problems such as lung damage, pnuemonia, etc. The first reported case of COVID-19 was discovered in Wuhan, China. The disease spread in Wuhan rapidly, and after that, the disease began to spread world wide at an alarming rate. The World Health Organization soon classified the disease as a pandemic.

The infection of COVID-19 in patients showed extreme variation in severity. In some cases, the patients showed extremely mild symptoms, while in other cases, they developed extremely severe infections. The infection severity was heavily impacted by the comorbidities of patients. Patients with comorbidities result in server infections as well as lead to fatal cases. Paitents with prior comorbidities such as diabetes, cardiovascular diseases, and respiratory problems showed higher severity.

Cardiovascular comorbidities include various circulation disorders such as arrhythmia, artery diseases, hypertension, and heart diseases. Arrhythmia itself is a disorder, but it also serves as a symptom in other disorders. An arrhythmia is an irregular beating of the heart. This leads to either a faster heartbeat (tachycardia), a slower heartbeat (bradycardia), or a random beating pattern. Detection of arrhythmia suggest the possibility of other cardiovascular disorders, hence it is primary test in patients admitted due to COVID-19 infections.

The electrocardiogram signals, or ECG signals for short, are used to prepare charts. These charts are used to detect arrhythmia in patients. These ECG charts are recordings of human heartbeats. The ECG is generally taken with an 8-lead or 12-lead system. Figure 1.1 shows the normal and abnormal ECG singals recorded on 12-leads. The heart rhythm is captured at a 120 Hz frequency. The chart is observed by a primary care physician. If the chart shows abnormalities, the patient is said to have arrhythmia. The patients with arrhythmias are further tested for other cardiovascular diseases or disorders.

The detection of arrhythmia from the ECG record is a slow and repetitive process. This process can be shortened by using a machine learning system. Use of machine learning systems will shorten the time required for detection of arrhythmia. This will also reduce the stress from
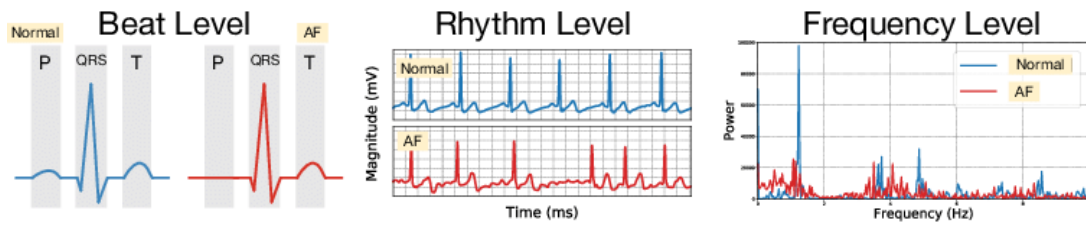
Figure 1.1: Normal ECG signal (Normal) and Abnormal ECG signal (AF) showing 1. Beat Levels 2. Rhythm Level 3. Frequency Level [13]

the medical supervisors and staff. As medical facilities have a large number of ECG records, machine learning system will be able to produce extremely accurate results. The use of machine learning will reduce human errors. This will allow the medical professional to focus on more complex problems and better patient care.

In this project, we are using a machine learning system to detect the arrhythmia in COVID-19 patients. The system takes the ECG records from the patients and generates a few models. These models are evaluated and compared against each other. The system employs a weightage-based selection system to select the most-suited model for the task.

1.1 Machine Learning

Machine learning (ML) is a subset of artificial intelligence (AI) technology. Machine learning is a process in which a computer learns from experience. The machine learning system is used to obtain meaning from the available data. Machine learning uses statistical models and advanced mathematics to find patterns and meaning from provided data. Machine learning does not depend on explicit programming but on learning algorithms derived from various statistical models. The objective of machine learning is to obtain meaningful experience and use this experience to solve complex classification and regression problems. The large amount of data and observations needed to generate a high performance machine learning model Another objective of machine learning is to allow the computer to learn from experience with minimum human interaction.

Machine learning is used in image and speech recognition, recommendation systems, prediction systems, chatbots, classification and detection systems. The use of machine learning is already prominent in the healthcare industry, smart cities, IT industries, R&D industries, etc. Machine learning takes the data from the users and converts it into structured or unstructured forms. While most algorithms and models prefer structured data, machine learning can utilize both structured and unstructured data according to its needs. For example, machine learning can forecast the

amount of solar radiation from various natural elements. Machine learning systems are generally classified into three types:

1.1.1 Supervised Learning

The supervised learning systems employ supervised learning algorithms for the learning. This system uses well-labeled data for training. The supervised machine learning algorithm maps the input data against the correct output provided with training data. Supervised algorithms use this learning approach to train themselves for classification and regression tasks.

The performance of supervised learning models is based on the input quality and dimensions of data. The subset of training data is used for the validation process, in which superparameters are trained to increase the performance of models. The supervised learning systems are further categorized into two subsets based on problems.

Classification Problems

Classification algorithms are a subset of supervised learning algorithms. The classification problems can be binary or multiclass problems. In binary classification, the output is primarily yes or no, true or false. But sometimes classifcation can be different depending on the type of problem. The classification algorithm analyzes the input variables and outputs the class based on the analysis.

Regression problems

Regression problems are solved by regressive supervised algorithms. In this problem, algorithms utilize the known relationship between input variables and output variables to train themselves. These algorithms are generally used in the case of forecasting, trend analysis, etc.

1.1.2 Unsupervised Learning

Unsupervised learning systems are machine learning techniques that employ unsupervised learning algorithms. These algorithms use unlabeled datasets for training purposes. The goal of an unsupervised system is to detect abnormalities in the dataset on its own. This makes the models better suited to unknown environments where they can learn information from any type of data. Unsupervised learning models work better than supervised models in the case of unstructured data as they derive meaning from data without needing human supervision. The advantage of unsupervised learning algorithms is that they are designed to mimic the way of

human thinking, making them autonomous and much closer to real artificial intelligence.

There are two major categories of unsupervised learning; they are clustering and assossiation. In the case of clustering, the data points from a dataset are grouped together based on training. These data points are grouped with their similarities to other data points present in the dataset. In the case of association learning systems, the relationship between multiple data points is identified or trends are identified. In other words, association algorithms obtain the relationship between various variables present in a dataset.

1.1.3 Reinforcement Learning

Reinforcement learning is a subset of supervised learning systems that utilizes feedback based learning systems. In this system, the machine is allowed partial human interaction to analyze its behavior. This system does not use labeled data, but it interacts with the environment on its own. This system is allowed to perform some actions in a given environment and learn from its own experience. This learning system is able to tackle extremely complex and difficult problems due to its dynamic learning approach. But the major disadvantage of this system is its cost-to-performance ratio as well as its longer training time.

Machine learning system is used in this dissertation to detect the arrhyhmia from ECG singals. The system is provided with a template of the five machine learning models, ECG records of the patients, and user preference parameters. The user parameters are used to generate the weightage. The data is split into 80:20 ratios. The system generates the five models using templates and 80% of the data. These five models are evaluated with weightage and the remaining 20% of the data.

1.2 Problem Statement

In current COVID-19 pandemic life threatening complications can be avoided with earlier diagnosis. One of such complications is arrhythmia which can be detected by reading ECG signals. This task is carried out by medical personnel. This repetitive and mundane task puts strain on medical staff.

Supervised learning models can be used for such tasks. These models have their own set advantages and disadvantages. Hence automating selection of the best model for required tasks will allow medical personnel to customise applications based on need. This will reduce strain from medical staff and allow them to focus on patient care.

1.3 Motivation

COVID-19 disease became a global pandemic in 2019. Since then, it has caused around 580 million infections world-wide and 6.42 million deaths, while in India it caused a total of 44.5 million infections, resulting in the deaths of half a million citizens. The infection severity varied from individual to individual, but it has been observed that patients with comorbidities were more likely to result in severe infections. The most common comorbidities observed in these patients were prior respiratory damage caused by other diseases or disorders. The second most common comorbidity was cardiovascular complications.

The cardiovascular comborbidities cause a large number of severe cases and fatalities in COVID-19 patients. The main contributing factor to this was the silent nature of these cardiovascular disorders. Most cardiovascular disorders are hard to detect and left untreated in a large number of cases. Earlier diagnosis of cardiovascular disease will lead to better treatment of patients. This will reduce the number of undetected cases. Cardiovascular diseases are often diagnosed by analyzing ECG records of patients for the presence of arrhythmia. This process is extremely time consuming. Automating this process will reduce the strain placed on medical staff significantly. This will allow them to tackle more complex and important problems. The early diagnosis of cardiovascular disease will allow the proper treatment and patient care for the patient.

During research, we noticed that there was a lack of literature for model selection systems based on user requirements. Therefore, another motivation behind the project is the development of an automated system for the selection of a model. In this system, models are selected from the weightage generated from user choices and performance evaluation of the models. The goal behind the weightage based system is to allow users to give input to the system based on the most important factors. This can be higher accuracy at the cost of prediction time, or lower accuracy at the cost of prediction time. This way, the users are given choice in the selection of the model that will meet their own requirements.

1.4 Beneficiaries

This system is primarily built for medical personnel tasked with the handling of COVID-19 cases. But it can be beneficial in the cases of patients who suffer from cardiovascular problems and other diseases.

The system is useful for a wide variety of tasks, such as detection of epilepsy or Parkinson's from brain wave charts, and detection of cancer and other disorders. This system will reduce the large

amount of workload from the medical staff. The system will help medical personnel get a better understanding of the patients' cases. This will result in better patient treatment and patient care.

Once the system is trained, it can be used for generalized predictions. On the other hand, this system can also be trained for specialized tasks. The system does not need machine learning experts to operate; i.e., it can be trained and used by end users directly. The lack of external connection will also increase security and privacy, while the system will maintain the high performance of the models.

1.5 Scope of the project

Although there are a large number of trained models and systems are present in machine learning, there is a lack of model selection systems targeted towards non-technical people. The aim of this dissertation is to propose an automated model selection system specifically built for non-technical people. This system accepts the user's requirements and provides the most suitable model for their needs.

The system takes a data set in a CSV format file, and the obtained results are returned in JSON format. These JSON files are stored for recording purposes and future use. The dataset provided can be of any dimension, and the data is processed by the pandas library. The dataset used in this process is obtained from the kaggel, and the models are trained with this dataset. The system provides the most suitable model for the prediction depending on the requirements provided by the user and the evaluation of the models.

1.6 Organization of Report

The report is structure into seven chapters as follows:

Chapter 1 - Introduction: This chapter gives an overview of the report. It gives a general overview of the current situation of COVID-19 pandemic and the stress it puts on the medical system. It also describes the proposed solution to tackle this problem.

Chapter 2 - Literature Survey: This chapter provides review of relevant literature of supervised learning systems and automation approaches for the model training.

Chapter 3 - System Architecture: This chapter describes the architecture of the system. It discusses the models used in the system, the automated training and selection of models.

Chapter 4 - System Interface: This chapter describes the user interface of the system. This chapter will discuss the pages accessible by the user and the information about the development server.

Chapter 5 - Result And Analysis: This chapter will describe the performance metrics used by the system for the selection process. The chapter also gives details about the dataset used for training and testing. It also provides recommended system specification required by the application.

Chapter 6 - Applications: This chapter gives various applications of the developed system.

Chapter 7 - Conclusion And Future Scope: This chapter concludes the report by providing a brief summary of the project. This chapter also describes the future scope of the system.

Chapter 2

Literature Survey

Chapter 2

Literature Survey

2.1 Introduction

COVID-19 has become a global pandemic in the past few years. It targeted the human respiratory system and caused serious infection in the patients. This infection affected patients quite severely in the case of the present comorbidities. This resulted in a fatal infection in some cases. These comorbidities were mainly diabetes, cardiovascular disease, prior respiratory system damage, etc. Cardiovascular diseases, also known as CVDs, are caused by disorders in the cardiovascular system, including the heart and other blood-circulating organs. Major Cardiovascular diseases are life-threatening, while minor Cardiovascular diseases can lead to heart attacks or strokes. The presence of this disease can lead to severe COVID-19 infection.

Cardiovascular disease is generally accompanied by various symptoms, but arrhythmia is the most common symptom among them. Babapoor-Farrokhran et al. (2020) suggest that arrhythmia is one of the most common symptoms in COVID-19 patients. An arrhythmia is an irregular beating of the heart. This can be easily detected by the ECG report from the patients. The ECG graphs are usually recorded at a 128 Hz frequency over a few minutes. These graphs are studied by the medical staff for the detection of arrhythmia. If any irregularity is present, the patient is said to have arrhythmia. This process can be easily automated by the machine learning system.

Babapoor-Farrokhran et al. also suggests that arrhythmia was present in about 7% of total COVID-19 cases in Wuhan and 14.8% of patients with poor outcomes had arrhythmia. The authors of Mulia et al. (2021); Liu et al. (2020) carried out seperate surveys of 17 studies consists of 5815 patients are concluded similar findings that up to 9.3% of COVID-19 confirmed patients had been detected with arrhythmia. The surveys Beri and Kotak (2020); Ren et al. (2020) showed that up to 94.4% of patients with arrhythmia resulted in fatal infection and 95.8% of patients with severe infections had arrhythmia. In the Yarmohammadi et al. (2021), author suggests that only 8% of patients with arrhythmia had prior cardiovascular disease, while in 56% of the cases, arrhythmia symptoms were new onset after contracting COVID-19 disease.

2.2 Machine Learning

Machine learning is becoming a widespread technology. Machine learning is a technology where the computers learn to make decisions based on provided data. Machine learning uses statistical

modeling and advanced mathematics to build decision making models.

The industries are making large amounts of data every day. Making sense of this data will give an advantage over the competitors. Machine learning technology is capable of this feat. Hence, many industries are already using machine learning. The medical industry is one of them. The medical industry uses machine learning for the research and detection of various diseases. Well trained machine learning models are extremely accurate and fast. This allows faster diagnosis, which leads to better patient care. Supervised machine learning and unsupervised learning are two main types of machine learning systems.

2.2.1 Supervised Machine Learning

Supervised machine learning is a subset of machine learning technology. The supervised learning system uses the labeled data to derive meaningful results. The computer learns the patterns from the data and uses labels to increase the accuracy of the models. The supervised models become more efficient with large amounts of data. Supervised learning models are generally used for classification and regression problems.

2.3 Review of related literature

2.3.1 Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison

Ağbulut, Gürel and Biçen (2021)

The authors used deep learning, SVM and KNN for calculation of daily global solar radiation. The radiation is directly proportional to the amount of solar power generated, hence correct assessment gives edge over the competition. The authors used various natural conditions as variables to correctly predict the amount of radiation per day. Authors suggested that neural networks are extremely accurate for their use cases.

2.3.2 A fast machine learning model for ECG-based heartbeat classification and arrhythmia detection

Alfaras, Soriano and Ortín (2019)

The authors used machine learning for the detection of the arrhythmia from ECG signals. The author used a single lead recording over a long period of time. The authors achieved up to 92.7% sensitivity and 86% accuracy in the first test, while 95.7% sensitivity and 75% accuracy in the second test. The authors utilized a GPU for the calculations to reduce the amount of training and prediction time. The authors suggest that good feature selection and guidelines are necessary for efficient machine learning models.

2.3.3 Automatic model selection for the optimization of SVM kernels

Ayat, Cheriet and Suen (2005)

The authors used novel criteria for model selection against the custom SVM classifier. The system needs prior knowledge of machine learning technology. The system was able to handle the data up to two parameters independently. The system was unable to handle multi class data, hence it should be partitioned appropriately. The automation system reduced the probability of errors.

2.3.4 Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results

Brazdil, Soares and Da Costa (2003)

The authors used a meta ranking approach for the selection of machine learning algorithms instead of the baseline approach. The models are ranked according to the features of the data. The methodology used for the ranking is based on the success rate of the model and time parameter. The author suggested that the study will be beneficial to develop future ranking methods.

2.3.5 A survey on the explainability of supervised machine learning

Burkart and Huber (2021)

The authors conducted a survey of the machine learning system in finance, healthcare and various other industries. This survey provided the overview of supervised learning algorithms. The

author suggests that the uncertainty about AI and machine learning is holding industrialization back. Also extremely complicated models are required for the complex industrial tasks making it harder to adopt. The survey suggested that the machine learning system was widely used during COVID-19 pandemic. Changing the approach towards the solution will increase the adaptation of machine learning. Also setting up clear ethical clauses in machine learning is necessary for wide use of this technology.

2.3.6 Machine learning in medicine

Deo (2015)

The author commented on the lack of meaningful research of machine learning in the medical field. The authors studied both supervised and unsupervised machine learning algorithms. The author concluded that the supervised learning methods are beneficial in risk assessment and detection of known diseases, while the unsupervised methods are beneficial in the same task in case of the novel diseases. The author suggested that the large amount of data is necessary for accurate predictions. The subject specific methods as well as wider support is necessary for wide scale use of machine learning in the medical field.

2.3.7 Machine learning in detection and classification of leukemia using smear blood images: a systematic review

Ghaderzadeh, Asadi, Hosseini, Bashash, Abolghasemi and Roshanpour (2021)

The authors reviewed studies conducted to detect leukemia from blood smears with machine learning. The authors suggest that machine learning eased the load on the personnel. The machine learning systems achieved up to 97% accuracy in leukemia detection in a case. Machine learning also produced more than 74% accuracy in other cases.

2.3.8 IoT for smart cities: Machine learning approaches in smart healthcare - A review

Ghazal, Hasan, Alshurideh, Alzoubi, Ahmad, Akbar, Al Kurdi and Akour (2021)

The authors used machine learning systems with IoT systems for smart cities. The authors suggested that sensors provide a large amount of data over time for the machine learning system. Authors suggest that machine learning will be extremely beneficial in the development of smart cities, advanced healthcare systems, and various other fields.

2.3.9 Battling COVID-19 using machine learning: A review

Chadaga, Prabhu, Vivekananda, Niranjana and Umakanth (2021)

The authors used a machine learning system for detection of viral disease. The machine learning system showed promising results. The authors are planning to do more research with various datasets to evaluate the efficiency and accuracy of machine learning.

2.3.10 The role of machine learning algorithms for diagnosing diseases

Ibrahim and Abdulazeez (2021)

The authors studied various machine learning techniques used for the diagnosis of diseases. The author suggests that machine learning reduces the number of human errors while making the system useful for daily life. The author suggests that KNN, SVM, DT, and RF models showed good results. The author also suggested that ensembling methods would produce even better results.

2.3.11 Classification of pachychoroid disease on ultrawide-field indocyanine green angiography using auto-machine learning platform

Kim, Lee, Park, Baek and Lee (2021)

The authors employed an auto machine learning platform to classify the pachychoroid disease. The authors used direct and indirect approaches for comparisons. The authors found that the indirect approach of equalizing the data produced better results. The author was satisfied with the auto machine learning algorithms, but suggested that the technology is still unstable for proper medical uses.

2.3.12 Automatic model selection for support vector machines

Lee and Lin (2000)

The authors proposed an automation system to generate the most-suited SVM models. Theory behind the research was the approximation of loo stopping rate was reducing the efficiency of the SVM models. The author suggested that better loo stopping will reduce the training and testing time. The authors used stalag collection after scaling appropriately for the model generation. The produced model is suitable for data upto 1000 variables. The system is currently limited to

SVM models and RBF kernels.

2.3.13 Deep transfer learning for industrial automation: a review and discussion of new techniques for data-driven machine learning

Maschler and Weyrich (2021)

The authors used transfer learning in the industrial tasks for industrial automation systems. The authors suggested that transfer learning and continuous learning technologies are important for rapid industrialization of machine learning technologies. These will be obtained by reducing the gap between research and practical uses of these technologies. Author also suggests that machine learning performs better in dynamic environments of industries.

2.3.14 Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset

Maseer, Yusof, Bahaman, Mostafa and Foozy (2021)

The authors used machine learning for the detection of anomalies. The DT, KNN, SVM and ANN algorithms were used for the tasks. The author also used some unsupervised learning algorithms, but supervised learning algorithms provided better results. The DT and KNN models provided bet results. The author concluded that supervised machine learning is still limited in case of complex and novel problems, but otherwise produced better results than unsupervised learning algorithms.

2.3.15 DDOS detection using machine learning technique

Pande, Khamparia, Gupta and Thanh (2021)

The authors used random forest methods for detection of DDOS attacks. Five properties of the DDOS attacks are used as the variable in machine learning algorithms. The authors suggested that the machine learning model random forest produced good results.

2.3.16 Classification of arrhythmia using machine learning techniques

Soman and Bobbie (2005)

The authors used machine learning to automatically classify the cardiac arrhythmias in embedded systems. The models were focused on three factors: accuracy, predictions and ease-of-use of the system. The author suggested that the volume of data is indirectly proportional to the understanding of the data. The designed model produced satisfactory results, hence proving the usefulness of machine learning in the medical field.

2.3.17 Auto machine learning-based modelling and prediction of excavation-induced tunnel displacement

Zhang, Shen, Huang and Xie (2022)

Authors proposed an AutoML system for the prediction of excavation of displacement. Six generic models were provided with seven input points and two properties. The system was implemented with 10-fold cross-validation methods to increase the efficiency. The models produced efficient and precise predictions. The authors concluded that AutoML is optimal for these predictive tasks. Authors also suggested that GA-MLP also provided satisfactory results.

2.3.18 Predicting reaction yields via supervised learning

Zuranski, Martinez Alvarado, Shields and Doyle (2021)

The authors used a machine learning system in case with a large number of features. Authors suggested that a similar result to traditional statistical methods was achieved with machine learning algorithms. The author suggested that good data collection techniques are necessary for better results. The authors suggested that use of machine learning methods are successful in the chemical domain. The author also suggests that more development is necessary to accommodate the vast amount of features in the chemical field.

2.4 Inference of Literature Review

The following inference can be drawn from the basis of literature review

- Most papers introduced here use supervised learning algorithms for the prediction. This technique needs a well labeled dataset for training and evaluation. Therefore, expertknowledge is needed during the training and evaluation process.

- Some papers concluded that it is possible to select suitable models with minimum human interactions automatically. The knowledge driven and data driven approaches are used for automated model suggestions.

- Most papers also showed that on-shelf models can be used for accurate and efficient use cases, but few authors used novel kernels and models to get greater accuracy and efficiency for predictions.

- The recent paper showed that there is great need for models that can be setup faster for required work. To solve this problem we can use limited data for training which is possible with supervised learning algorithms according to few studies.

- Finally it can be inferred that a very negligible amount of work has been done for the automated model selection for general prediction purposes. Moreover, a limited amount of study is done with the general population as end users. Hence, this system is focusing on training, evaluation and selection of models for users with limited knowledge of data science.

Chapter 3

System Architecture

Chapter 3

System Architecture

Use of machine learning for the diagnosis of diseases has become common practice in recent years. The healthcare system uses machine learning systems for both patient care and research purposes. Supervised learning systems are used for the detection of already known diseases, while unsupervised learning systems are employed to understand novel diseases.

These systems make use of patients charts, records, and scans to provide a correct diagnosis. The proposed systems use the patients ECG charts to train various machine learning models. The system primarily provides two services: generating the most suitable model and providing a prediction system for medical staff. Figure 3.1 shows how these three modules work in a modular format.
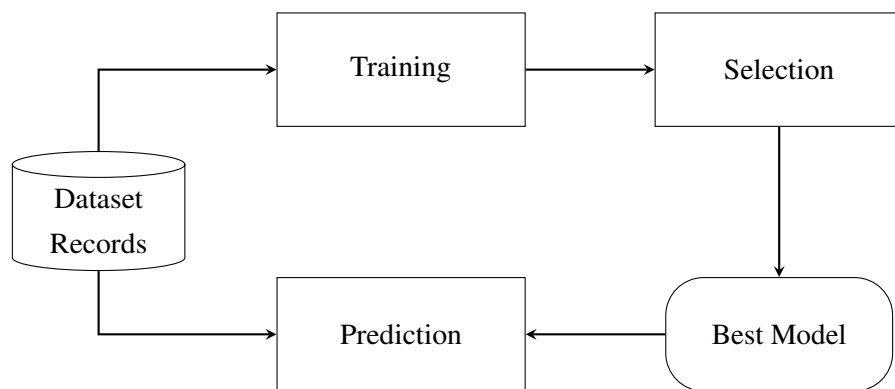

Figure 3.1: System architecture

The trainer module is responsible for both the generation of the models and the training process of the models. This module generates the model from the predefined machine learning templates. These templates are tuned for the detection of anomalies in frequency. All models are trained and stored in the temporary folders for the evaluation process. The performance of the models is evaluated using the predefined metrics.

The selection module is responsible for the Vscore calculation. The weightage is generated from the user choices and used in this process. The performance score obtained from the previous module is used along with this weightage to find the weighted sum of the performance parameters. The model with the highest Vscore is selected as the most suitable model. The Vscore of the model is highly dependent on the user preferences, specifically the time parameter.

The most suitable model is stored for future use in a specific directory; this can be accessed by the dashboard and prediction module. The prediction module also provided two services: prediction for multiple patients and prediction for single patients. The prediction system lists the most suitable models for the users. The user is allowed to select one of the models for their needs.

3.1 Data Flow in system
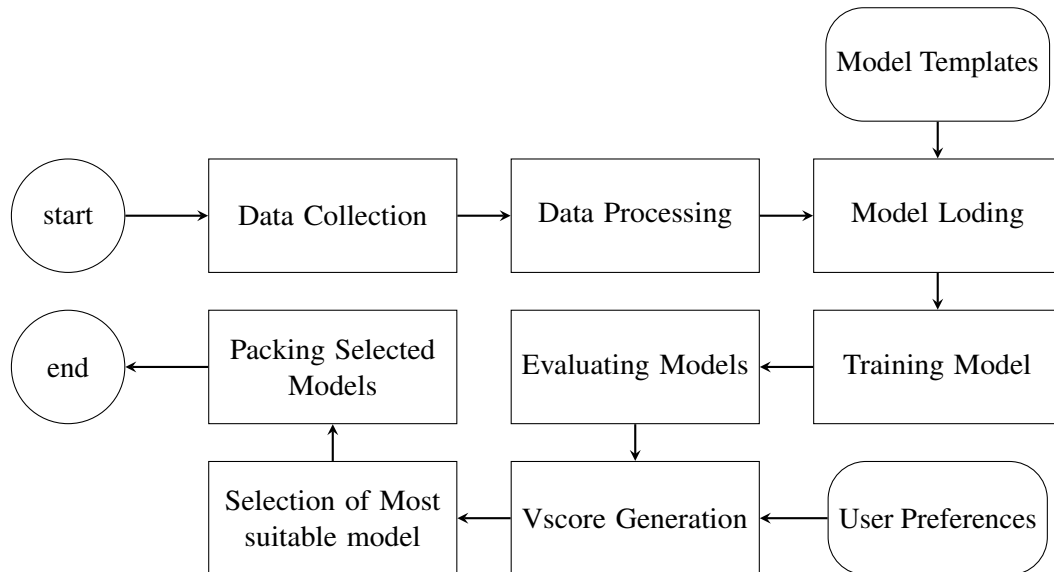
Figure 3.2: System Architecture - Selection Process

The data used in this process goes through this sequential process shown in fig. 3.3

Figure 3.3: Training and Selection Process

Data collection: Collection of data is the first step in this process. The data is collected from the user. The data is stored in a local directory for further processing. The data is then divided into training and testing sets.

Data processing: In this step data labels are converted to 1 or 0, with respect to previous values.

In case of data with more than 2 labels, labels with min value are converted to 0 and other values are converted into 1.

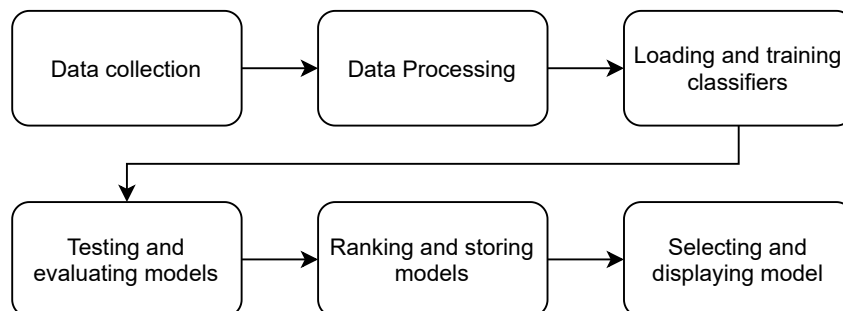**Loading and training classifiers:** In this step model templates are used for generation of models. These models are trained with the help of processed training data.

Testing and evaluating models: The trained models are evaluated with help of processed testing datasets. The performance metrics are stored for ranking models.

Ranking and storing models: Performance metrics obtained in previous steps are used to rank the models, the models are stored in a local directory.

Selecting and displaying model: In this step rank of models are used for selection of best model, this model is stored into best models directory for future predictions. The name of this selected model is displayed to the user.

3.2 Training and selection process

As shown in fig. 3.4, the system is provided with a dataset, in this case records of ECG reports of patients. This data is already provided with labels. These labels can be boolean values i.e., 1 for True, and 0 for False, or they can have a series of values starting from 0. This data is further divided into training dataset and testing dataset. These datasets are forwarded to the extraction process. In the extraction process, datasets labels are converted into 0 and 1, this will eliminate extra classes present in labels by changing their value to 1. The processed training dataset is sent for the model training. As soon as this process takes place, the training module accesses premade models templates, and generates the model for training purposes. These models are trained with processed training data and stored in a directory for evaluation purposes.
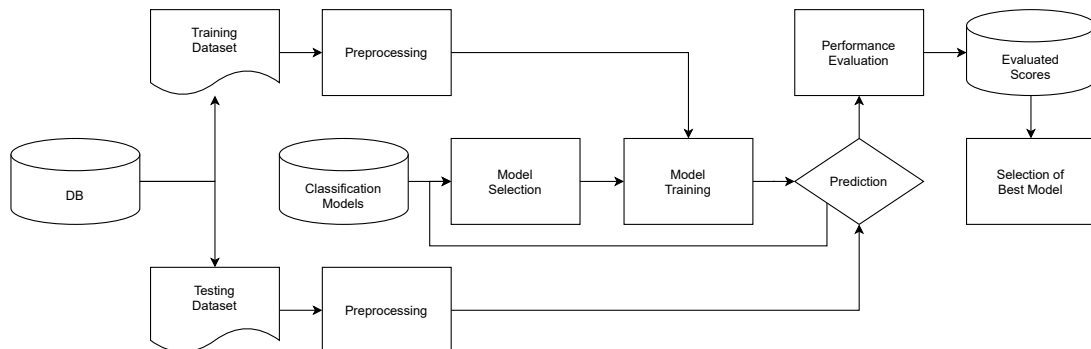

Figure 3.4: Training and Selection Process

Selector module accesses the stored models and processed testing dataset to evaluate the

performance of models. The performance metrics used for evaluation are accuracy score, F1 score, precision score, recall score, roc score and prediction time of model. These parameters are multiplied with default or user provided weightage provided to generate scores of the models. These scores are stored and used to select the best suited model.

3.3 Methodology

Figure 3.5 shows the approach to the selection of a suitable model. The system can handle an infinite number of models. For each model, six performance parameters are used. These performance parameters are accuracy P_1, F1 score P_2, precision P_3, recall P_4, area under the ROC P_5, and prediction time P_6 (further explained in section 5.1). From these, the first five parameters are grouped and given a weightage range between 0.2 and 1.0, whereas prediction time is given a weightage range between 0.25 to 0.75. Equation (3.1) shows the mathematical formula used to calculate the Vscore of the models.

For models M_1, M_2, ..., M_n, the Vscores V_1, V_2,..., V_n are obtained. These obtained Vscores are compared with each other. The model with the highest Vscore is selected as the most-suited model.

$$V_{score} = \left(\sum_{x=1}^{5} w_x P_x \right) - w_6^2 P_6 \tag{3.1}$$

where,

V_{score} Vscore of the model

w_x The weightage generated by the system for the xth parameter.

P_x Performance of xth parameter

3.4 Classifiers Used

The classifiers are algorithms that classify data into two or more classes based on rules set out by users. The models trained with these classifiers are used for classification purposes. In this project we are using classifiers to train models, these models will output data into two classes.

By default this system uses five different classifiers, they are K-Nearest Neighbors, Decision Tree, Random Forest, Multilayer Perceptron and Support vector machines. These classifiers are
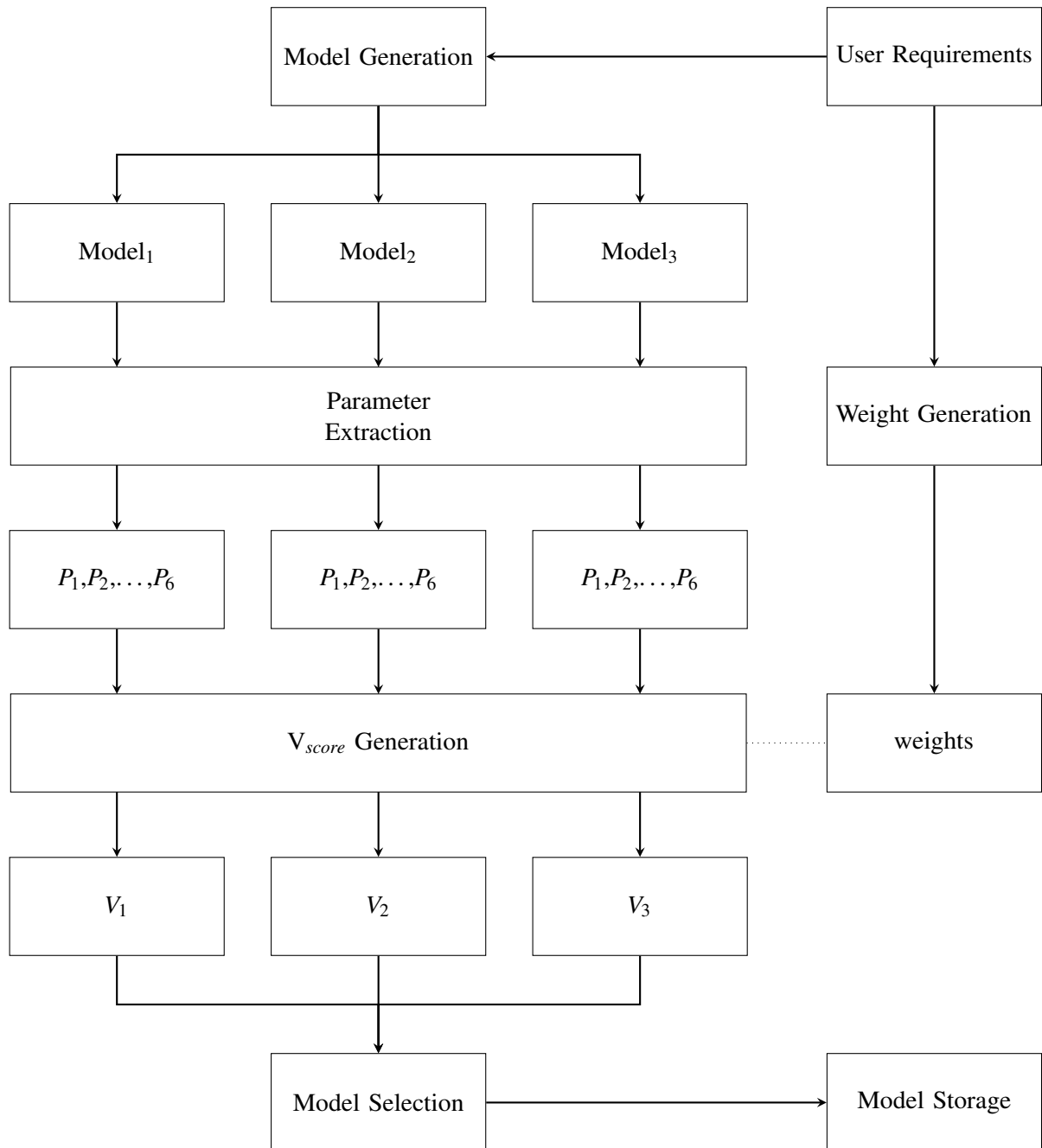
Figure 3.5: Model Selection Approach

categorized as supervised learning algorithms or classifiers. These classifiers will predict the classes of data provided by the user.

3.4.1 K-Nearest Neighbors

K-nearest neighbor (KNN) algorithm is a supervised machine learning algorithm. Classification and regression problems such as multiclass problems are solved using this algorithm. KNN is lazy learning and non-parametric algorithm, which can be advantages or disadvantages depending on the user's requirements. These properties make this algorithm very effective and uncomplicated in general tasks. The drawback of the simplicity is the slow prediction time.

Algorithm for K-Nearest Neighbors:

Step 1: Determine the number of K neighbors.

Step 2: Determine the Euclidean distance between the K neighbors.

Step 3: Using the estimated Euclidean distance, find the K nearest neighbors.

Step 4: Count the number of data points in each category among these K neighbors.

Step 5: Assign new data points to the category with the greatest number of neighbors.

Step 6: The model is now complete.

3.4.2 Decision Tree

Decision trees are the most popular and effective tool for classification and prediction. The decision tree is a flow chart-like tree structure, where each internal node specifies a test for the attribute, each branch represents the result of the test, and each leaf node contains a class label. Decision trees are the most popular high-performance tool for classification and prediction.

Construction of Decision Trees:

The trees are trained by subdividing the features into subsets based on the attribute values. This process is run recursively on derived subsets. The recursion completes when all subsets of nodes have the same value for the target variable or if the split doesn't add any additional value to the prediction. Building a decision tree classifier is suitable for exploratory knowledge discovery as it does not require domain knowledge or parameter adjustment. Decision trees can handle high-dimensional data. In general, decision tree classifiers are more accurate. Decision tree induction is a typical inductive approach to learning classification knowledge.

Alogrithm for Decision Trees:

Step 1: Start the tree on the root node that contains the complete dataset.

Step 2: Use the Attribute Selection Scale (ASM) to find the best attribute in the dataset.

Step 3: Divide S into a subset containing the possible values of the best attributes.

Step 4: Create a decision tree node with the best attributes.

Step 5: Recursively build a new decision tree using the subset of the dataset created in step 3. Continue this process until you reach a stage where you can no longer classify the node, and you can call the last node a leaf node.

3.4.3 Random Forest

Random forest is a supervised learning algorithm based on a decision tree algorithm. This algorithm generates multiple decision trees from the provided sample. These decision trees are combined, and a majority vote is taken to solve the classification and regression problem. The Random Forest classifier offers more accurate and stable predictions at a slightly slower prediction time when compared with a decision tree.

Bagging

The random forest classifier is a type of ensemble classifier which uses the bagging technique. Random samples from data sets are collected. These samples generate independent models. Random forest uses decision tree classifiers for model generation. The final output is generated based on majority voting after combining the results of all models. This process of combination and majority voting is known as the aggregation process.

Classification algorithm for Random Forest:

Step 1: N number of random records are taken from the dataset with k number of reports.

Step 2: Each sample generates a separate decision tree.

Step 3: Each decision tree will generate an output/prediction unrelated to other trees.

Step 4: The final output is determined based on majority voting in classification problems.

3.4.4 Multi-layer Perceptron

The multilayer perceptron is a neural network-based supervised learning algorithm. In this network, input and output are not linearly mapped. It is a feedforward algorithm hence more suitable for classification and regression problems. MLP uses a backpropagating algorithm for training, making it faster and usable for general predictions.

Construction of MLP

The neural network is made up of three primary components, neuron, activation function, and layers.

Neuron: A neuron is the smallest unit of a neural network. It is based on neurons in the brain, which take one or more inputs to produce an output. The inputs are weighted, and each neuron has biases which are also weighted similar to inputs.

Activation functions: The weighted sum of inputs and bias is passed through activation functions. This function serves as a mapping between two layers. These functions are usually non-linear. Activation functions control the firing of a neuron. We are using the sigmoid as an activation function. The sigmoid function returns either a zero or postitive value.

Layers: Layers are rows of neurons in the network. The input layer is the entry point for the neural network. The output layer serves as the exit point for the neural network. A hidden layer or series of hidden layers connects input and output layers. In MLP, input layers take multiple features as data. The output layer returns either 1 or 0.

Algorithm for Multilayer Perceptron:

Step 1: N number of epochs is calculated by dividing sample size S by batch size B.

Step 2: Input layers accept the data features.

Step 3: Input layer forward data to hidden layers, which forward data to the output layer.

Step 4: Errors are calculated by comparing expected output and network output.

Step 5: The error propagated backward to adjust the weights of the network.

Step 6: All weights in the network are adjusted.

Step 7: Process in steps 3-6 is known as an epoch. The process is run for N number of epochs.

3.4.5 Support Vector Machine

Support Vector Machines or SVM is one of the most commonly used supervised learning algorithms for data classification and regression. This algorithm is primarily used to solve classification problems in machine learning. The SVM algorithm created decision boundaries to divide n-dimensional spaces into classes. The new data points are categorized using these classes. The best boundary line which compartmentalizes most data accurately is called a hyperplane. The SVM algorithm uses vectors to create the hyperplane. Support vectors handle extreme cases giving the algorithm its name Support Vector Machine.

Hyperplane: A decision boundary separates new data points into different classes. The decision boundary that produces the most accurate categorization is called the hyperplane. The hyperplane

adjusts with new data points by repositioning or introducing new dimensions. Hyperplanes can be linear or non-linear, depending on user requirements.

Support Vectors: These are extreme data points present in data sets. Support vectors have very close proximity to the hyperplane. Support vectors can directly affect the position and dimensions of the hyperplane.

Tuning parameters of SVM

Tuning parameters used in this projects are:

Kernal: SVM uses the kernel function to solve problems. These functions allow smoother operations on high dimensions. With kernels, SVM hyperplanes can theoretically reach infinite dimensions. Kernels also reduce complexity at higher dimensions. The primary purpose of the kernel is to provide a hyperplane way to accommodate non-linear datasets. In this project, we are providing an RBF kernel. This kernel is suitable for both small and large quantities of data.

C parameter: This is also known as the regularization parameter. C parameter suggests SVM optimization amount of margin used by a hyperplane. A higher value of the C parameter produces accurate results. Conversely, a lower value results in fast but less accurate predictions.

Gamma: The Gamma parameter defines the amount of influence of a model over the dataset. Gamma is inversely proportional to the curve of influence of the model. Lower gamma results in higher accuracy, but it is prone to overfitting. Higher gamma is constraining but can't handle complexity.

Chapter 4

System Interface
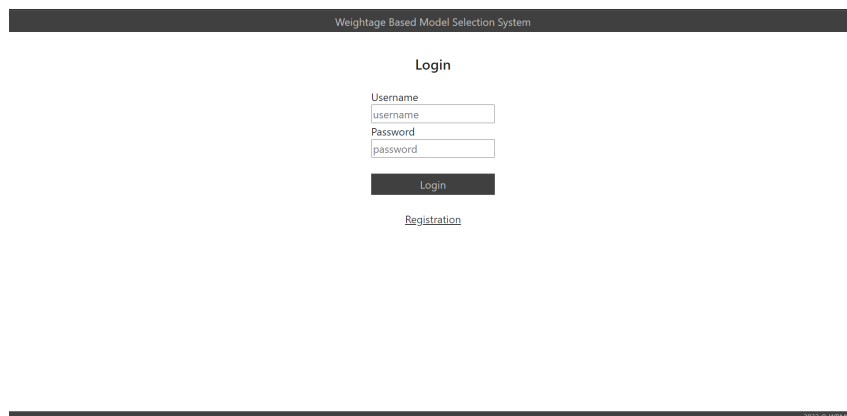
Chapter 4

System Interface

The system takes data in CSV format. A web-based GUI will be offered to users. We are using a flask-based web app for easier integration of the main system. We have implemented separate pages for users depending on tasks. These pages are the selection page, prediction page, performance page, and cross-performance page. We will see a detailed explanation of the pages and the process carried out in the background.

4.1 System Design

The server is accessed with the help of a port number. The site is accessed by URL in the form of `http://ip_address:port_number` URL syntax. In this development server we are using `http` protocol and `127.0.0.1` or `localhost` as `ip_address`. The `port_number` 5000 is used as per flask recommendation. Hence, the URL used by the application to access the website is `http://localhost:5000`. In the next subsections, we will go through each page available to the user.

4.1.1 Login Page

This page is the first page encountered by a user when logging on to the website. This page will take a valid username and password from the user and start the session for that username. After starting the session, the user will be redirected to the Home page. Figure 4.1 displays the login page.

Figure 4.1: Login Page

4.1.2 Home Page

The home page is the landing page for the users after the session is started. It provides information about the website. The side navigation is provided for users for easier access to the rest of the website. This side navigation is present on all the pages except the login page.
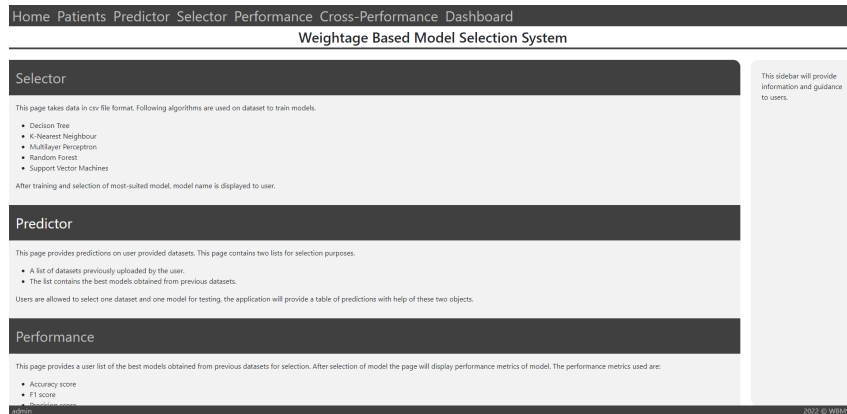
Figure 4.2: Home Page

4.1.3 Selector Page

This page takes file input from the user. Currently, only CSV file format is supported. The files are copied into the directory of the application. The training and selection module will process the copied file. These modules will execute model training, performance evaluation, and selection process. After process completion, the user will receive confirmation. Figure 4.3 displays the selector page.
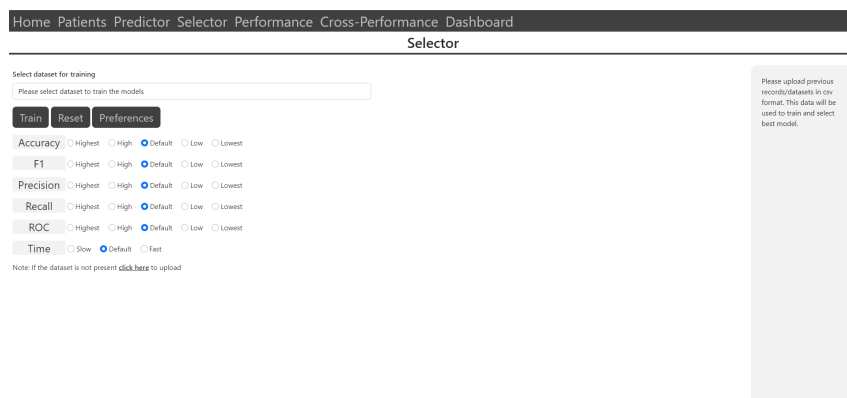
Figure 4.3: Selector Page

Display Selected Model Page

This page is a subpage to the selector page. This page is used to display the name of the selected classifier to the user. This page has a similar layout to the selector page with a small difference, and it acts similarly to the selector page. Figure 4.4 displays the selector-upload subpage.
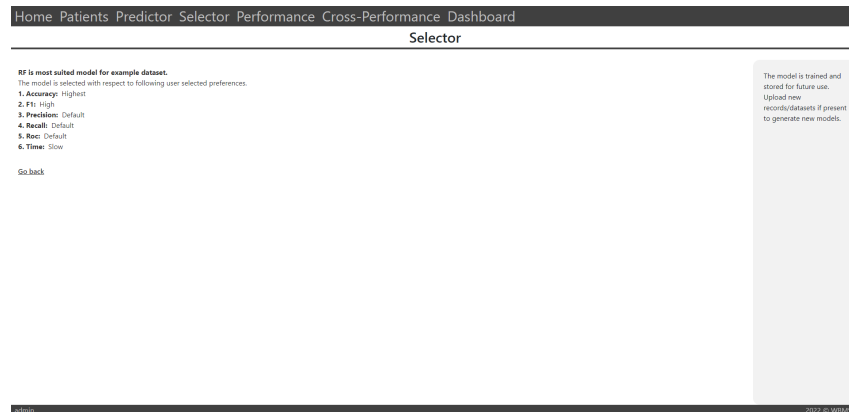
Figure 4.4: Display Selected Model Page

4.1.4 Patients Page

This consist of two subpages 1. registration page and 2. reports page. The registartion page generates user registration form and accepts values from logged user. Figure 4.5 (a) shows the registration form. The reports page provides user a list of patinets, as shown in fig. 4.5 (b). The result of selected patient is displayed along with link to his ECG report, as shown in fig. 4.5 (c).

(a) Patient Registration Page

(b) Patient Selection Page

(c) Patients Report

Figure 4.5: Patients Pages

4.1.5 Predictor Page

This page is used to make predictions based on the dataset provided by the user. This page generates a list of the best models for users to select models for predictions. The page also generated a list of datasets provided by the user for predictions. If the dataset is not available, the user is allowed to upload the dataset from the page. After submitting the selected model and dataset page will run the prediction process on the datasets. After completion of this process, the user is forwarded to the subpage. Figure 4.6 displays the predictor page.

Figure 4.6: Predictor Page

Predictor Results Page

This page will display the results of predictions obtained from the model and dataset submitted by the user with the predictor page. The result is displayed in tabular format with ID and prediction results as columns. Figure 4.7 displays the predictor-results subpage.

Figure 4.7: Predictor Results Page

4.1.6 Performance Page

This page allows users to select models from a list of models created from previous datasets. After selecting the model, the page will get performance data stored in the local directory and redirect the user to the subpage. Figure 4.8 displays the performance page.
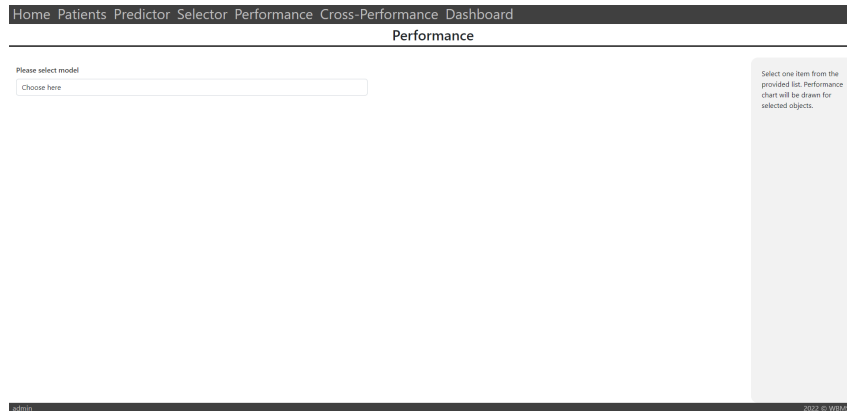
Figure 4.8: Performance Page

Performance Display Page

This page will display the performance obtained from the performance page. This page has a similar base layout to the performance page and has an option to display the graph generated from performance. This page also acts similar to the performance page. It allows the user to select another model to display its performance. Figure 4.9 displays the performance-display page.
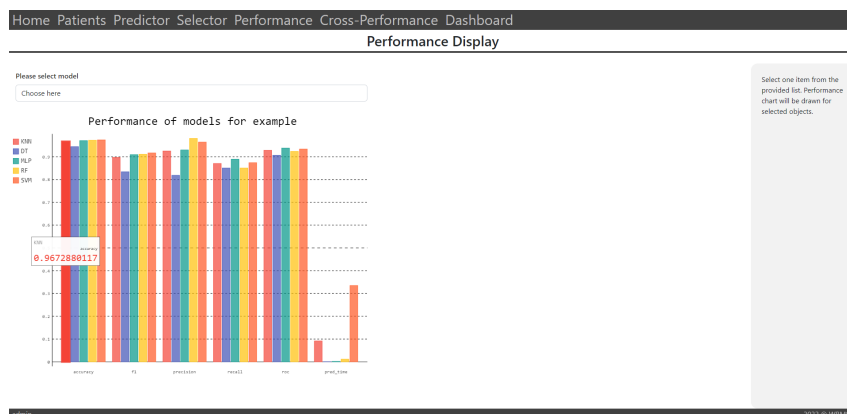
Figure 4.9: Performance Display Page

4.1.7 Cross Performance Page

This page allows users to select models from a list of models created from previous datasets. The page also provides a list of datasets uploaded by the user for predictions. The user is required to select a dataset from this list. The user-selected dataset and model will be sent to the server for cross-performance analysis. After completion of the cross-performance process, the user is redirected to the subpage. Figure 4.10 displays cross-performance page.

Figure 4.10: Cross Performance Page

Cross Performance Display Page

This page will receive the data obtained from the cross-performance process and display the data in chart format. This page has a similar layout and acts similar to a cross-performance page, with a display place for charts. Figure 4.11 displays the cross-performance-display page.

Figure 4.11: Cross Performance Display Page

4.1.8 Helper Pages

These pages act as helper pages for users as well as handles errors. In the instance of unauthorized access or invalid information, users are redirected to these pages. Figure 4.12 shows the error pages.

(a) Unauthorized Access Error Page

(b) Not Found Error Page
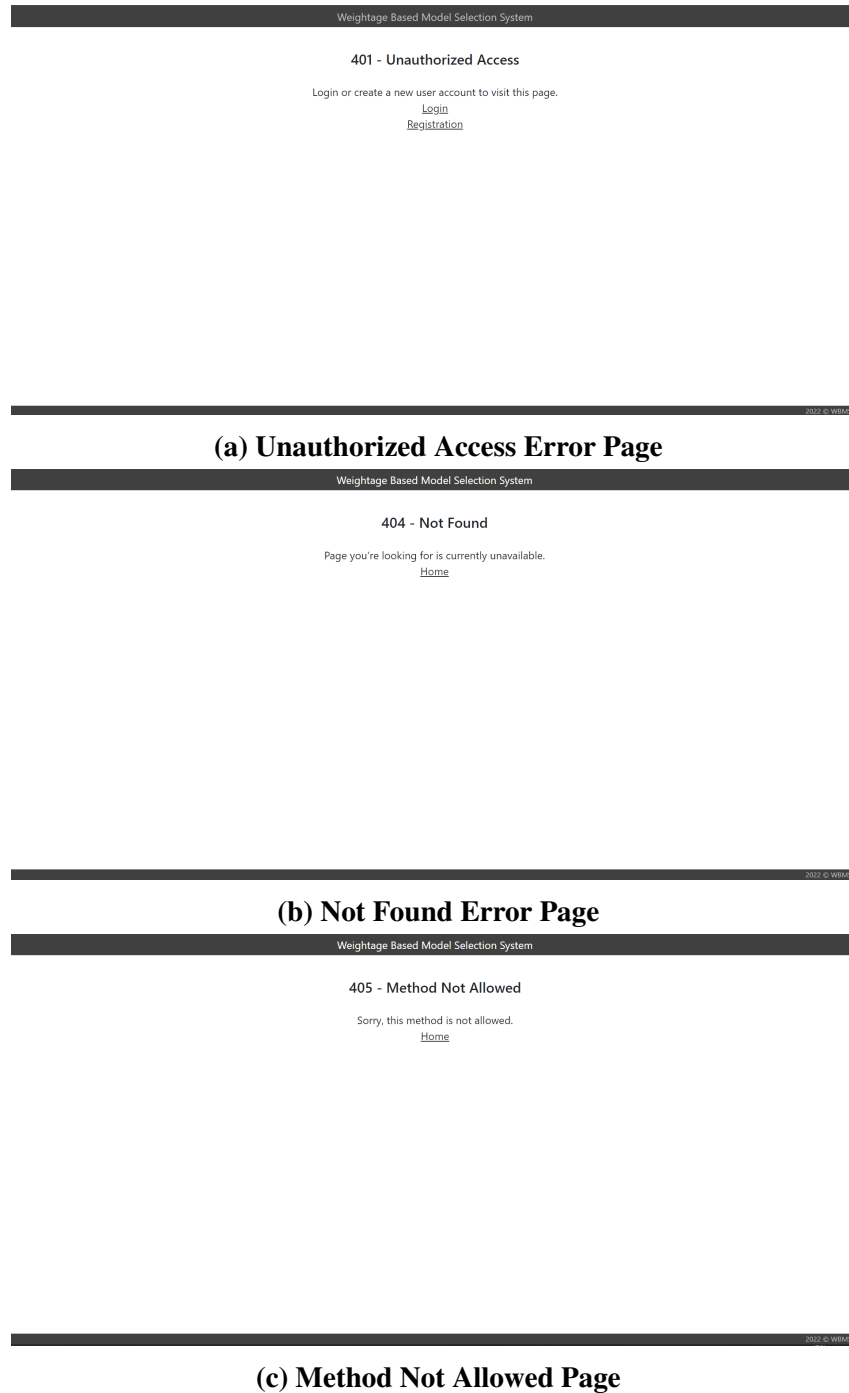
(c) Method Not Allowed Page

Figure 4.12: Error Pages

4.2 Hardware Details

The hardware used for the program is given below. The system needs at least a moderate amount of memory for operations. The amount of system memory used during development is 16 GB. The system also needs a high-performance CPU for the model training. The number of cores available for calculation is important too. The development system uses a 3.6 GHz processor with 6 Cores. The data uploaded by users and system-generated data need storage space. The recommended size of the storage is 250 GB.

Standard input and output devices are required to take data and commands from users and display them.

Table 4.1: Hardware Details

Hardware	Details
Installed memory (RAM)	16.00 GB
Storage	250 GB SSD
Processor	AMD Ryzen 5 3500 6-Core Processor 3.6 GHz
GPU	NVIDIA GeForce GTX 1650 Super
Input device	Standard Keyboard and Mouse

4.3 Software Details

The software requirements of the application are described below. The application is developed on Windows 10 system, but it is OS independent application. The technologies used are python as a base programming language, sklearn library to make model templates, and flask to develop the web application. Pandas library is used for data handling. HTML, CSS, and JavaScript are used for web pages.

Table 4.2: Software Requirement

Software	Details
Operating System	Windows 10
Technology	Python, Scikit-learn, Pandas, Flask, HTML, JavaScript

Chapter 5

Result And Analysis

Chapter 5

Result And Analysis

5.1 Performance Measures

There are a lot of different types of parameters to evaluate the performance of machine learning models. In case of the classification models Accuracy score, F1 score, Recall score, Precision score and total area under ROC curve used for performance evaluation.

5.1.1 Accuracy Score

The accuracy score is a fraction of the correct prediction by model with respect to total predictions by model. It can be represented by following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

5.1.2 Precision score

The precision score is a fraction of the correct positive predictions with respect to all positive predictions of the model. Higher precision scores result in fewer false positive predictions. It can be represented by following formula:

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

5.1.3 Recall score

The recall score is the fraction of correct positive predictions with respect to all predictions of the class. It can be represented by following formula:

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

5.1.4 F1 score

The F1 score is the weighted average of the recall score and precision score of the model. F1 scores are more reliable than accuracy scores in case of biased or uneven dataset. It can be represented by following formula:

$$F1 = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \tag{5.4}$$

F1 score can also be represented as $F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$

5.1.5 AUC ROC score

The ROC is a classifier's predictive quality that compares and visualizes the trade-off between the model's sensitivity and specificity. In graphical format, the area under it gives a relationship between false positives and true positives. The higher these areas are, the better the predictive quality of the model.

5.1.6 Prediction Time

Prediction times are nothing but the amount of time required by the classifier to make predictions for certain testing datasets. A model with a lower prediction time is desirable.

5.1.7 Performance Evaluation Methodology

The performance evaluation of the model is carried out with the help of Vscores. These Vscores are calculated with the help of performance scores and system generated weightage. The performance scores of models are calculated from the prebuilt libraries. The weights are calculated from the user-based choices. The preference provided by the user is used to generate the significance of a metric. The metrics are assigned the appropriate weightage based on their significance.

Perforamance scores and weightage are fed into the eq. (3.1) to generate Vscores. The model with the highest Vscore is selected as the most-suited model.

5.2 Dataset Description

The goal of the project is to detect the presence of arrhythmia from ECG signals accurately and faster than the traditional approach. For this task, a well-known open-access database MIT-BIH Arrhythmia Training and Testing Dataset are used 10. Both datasets are divided into four equal parts randomly and used for training and testing respectively. Each training set contains 21888 signals and the testing set contains 5473 signals.

5.3 Performance Evaluation

The datasets obtained from the MIT-BIH training dataset showed significant similarities in performance evaluation. All four datasets showed higher performance metrics for SVM classifiers. The prediction time required by SVM and KNN was significantly higher than other classifiers. The desired result can be tweaking ranking parameters. The performance of models is shown in fig. 5.1.

Table 5.1 shows the Random Forest model have highest Vscore for dataset 1, hence it is selected as the most-suited model for this dataset. While tables 5.2 to 5.4 shows the Support Vector Machine model scored higher Vscore for other three datasets, hence it has been chosen as most-suited model for those datasets. Table 5.5 shows the V_{score} of all models.

Table 5.1: Performance of models trained on Dataset 1

Metric	KNN	DT	MLP	RF	SVM
Accuracy	96.72	94.46	96.89	**97.33**	97.40
F1	89.71	83.43	90.05	**91.30**	91.69
Precision	92.53	81.86	94.71	**97.95**	96.43
Recall	87.06	85.06	85.84	**85.50**	87.40
ROC	92.84	90.68	92.45	**92.57**	93.38
Time(s)	0.457	0.001	0.002	**0.015**	0.297
$\mathbf{V}_{score}$	2.747	2.643	2.782	**2.807**	2.482

Table 5.2: Performance of models trained on Dataset 2

Metric	KNN	DT	MLP	RF	SVM
Accuracy	96.83	95.04	96.69	97.09	**97.46**
F1	90.28	85.50	89.73	90.75	**92.15**
Precision	94.25	84.90	94.73	98.60	**96.79**
Recall	86.63	86.09	85.23	84.05	**87.93**
ROC	92.77	91.48	92.13	91.90	**93.66**
Time(s)	0.435	0.001	0.003	0.014	**0.295**
$\mathbf{V}_{score}$	2.758	2.684	2.772	2.795	**2.808**

Table 5.3: Performance of models trained on Dataset 3

Metric	KNN	DT	MLP	RF	SVM
Accuracy	97.07	94.64	96.41	97.22	**97.44**
F1	90.93	84.30	89.34	91.21	**92.00**
Precision	95.82	83.81	90.13	98.37	**97.81**
Recall	86.53	84.80	88.57	85.02	**86.85**
ROC	92.87	90.73	93.29	92.36	**93.22**
Time(s)	0.404	0.001	0.002	0.017	**0.293**
$\mathbf{V}_{score}$	2.774	2.658	2.766	2.803	**2.803**

Table 5.4: Performance of models trained on Dataset 4

Metric	KNN	DT	MLP	RF	SVM
Accuracy	96.84	94.99	95.34	96.88	**97.15**
F1	90.52	85.73	85.01	90.37	**91.39**
Precision	95.71	85.91	97.83	98.65	**97.41**
Recall	85.86	85.55	75.15	83.37	**86.07**
ROC	92.52	91.28	87.40	91.56	**92.79**
Time(s)	0.452	0.001	0.002	0.014	**0.294**
$\mathbf{V}_{score}$	2.760	2.687	2.674	2.786	**2.790**

Table 5.5: V$_{score}$ of all models

Models	Training Dataset			
	Dataset 1	**Dataset 2**	**Dataset 3**	**Dataset 4**
KNN	2.747	2.758	2.774	2.760
DT	2.643	2.684	2.658	2.687
MLP	2.782	2771	2.766	2.674
RF	**2.807**	2.795	2.803	2.786
SVM	2.482	**2.808**	**2.803**	**2.790**
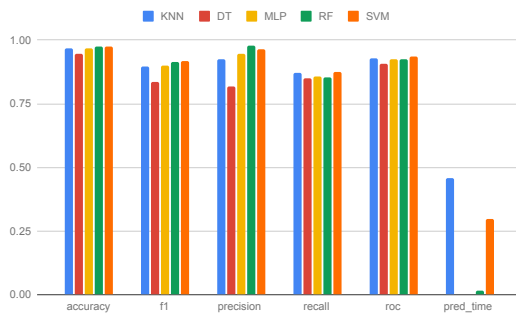
(a) Dataset 1
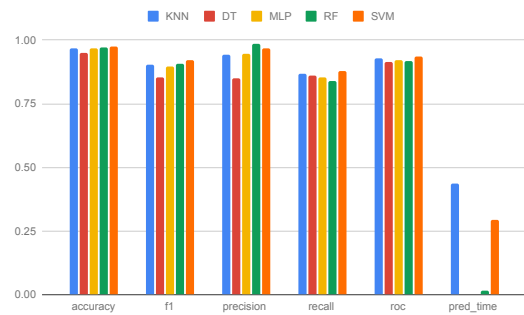
(b) Dataset 2
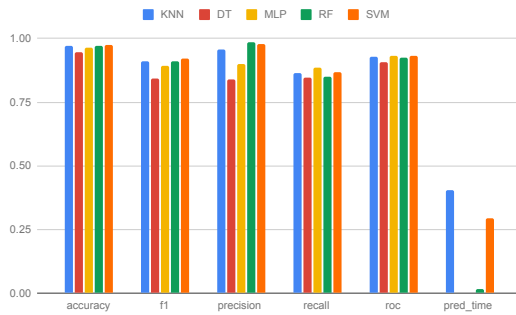
(c) Dataset 3
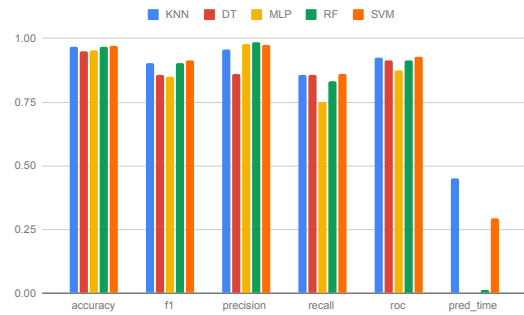
(d) Dataset 4

Figure 5.1: Perfromance results

5.4 Cross Performance Evaluation

Models tested against other datasets show a slight difference in performance. This difference suggests that the models are capable of performing general predictions. These general tasks can be performed with a small efficiency cost. Figure 5.2 shows the performance of models tested on all training datasets.
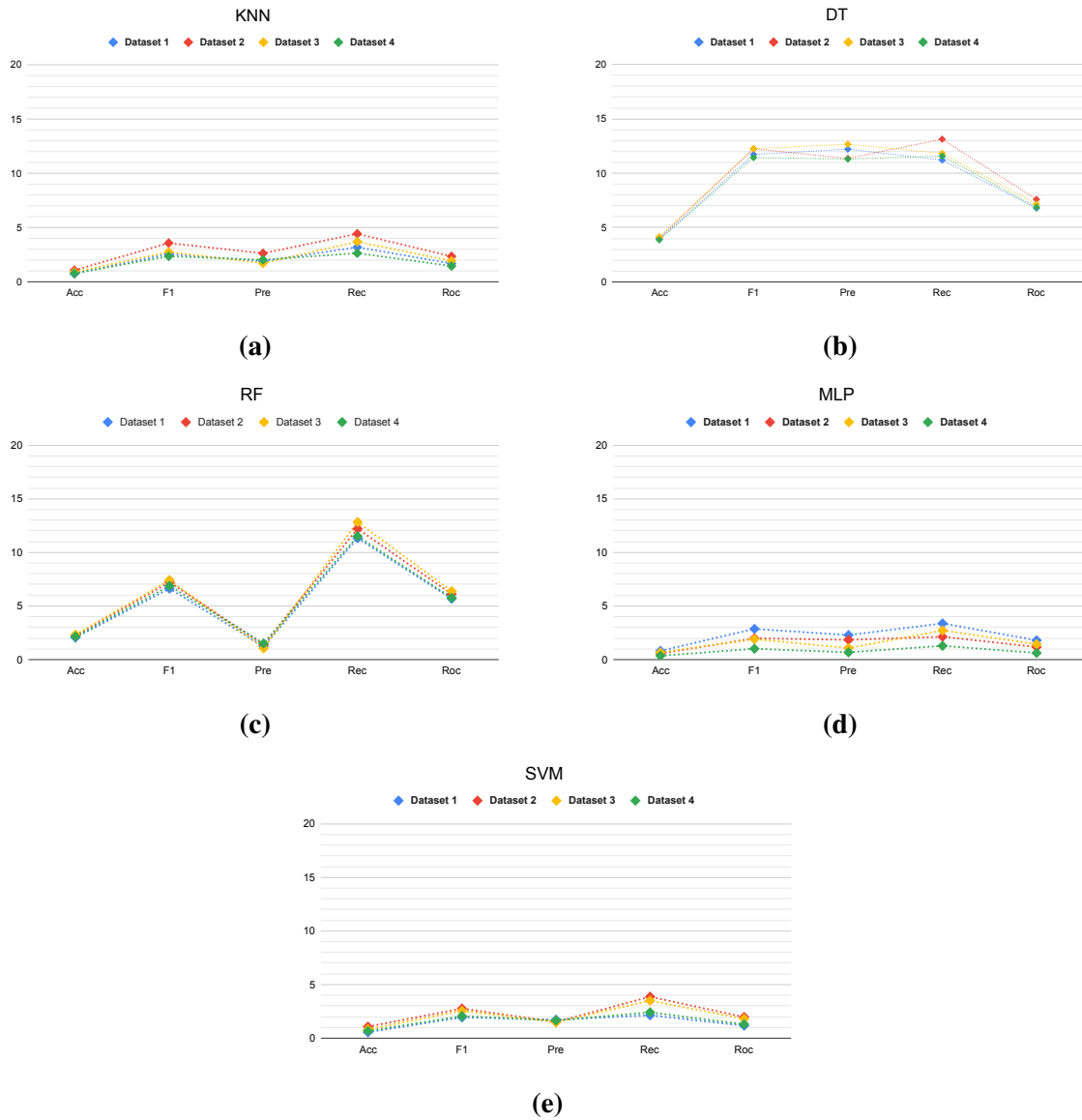
Figure 5.2: Average perfromance errors

The fig. 5.2 shows the average performance difference of models concerning training datasets. From fig. 5.2 (b) and fig. 5.2 (c), we can see that the difference in performance metrics of the Decision Tree and Random Forest algorithm is almost similar to all training sets. It indicates that these models can be used as solutions to similar problems. Figure 5.2 (d) shows the maximum difference in performance metrics when tested against other datasets. It indicates that the model is overfitting and is not suitable to be used for other datasets. While fig. 5.2 (a) and fig. 5.2 (e) shows good performance metrics for some datasets. This shows models can be tuned for specific requirements. This will produce more satisfactory results in general predictions. Tables 5.6 to 5.10 shows the cross-performance evaluation of models.

Table 5.6: DT model cross-performance results

Dataset	1	2	3	4
Accuracy	0.98	0.94	0.94	0.94
F1	0.96	0.85	0.85	0.84
Precision	0.95	0.83	0.84	0.83
Recall	0.96	0.86	0.85	0.85
ROC	0.97	0.91	0.91	0.90

(a) Dataset 1 - DT Model

Dataset	1	2	3	4
Accuracy	0.94	0.98	0.94	0.94
F1	0.84	0.96	0.84	0.85
Precision	0.85	0.96	0.85	0.85
Recall	0.82	0.96	0.84	0.84
ROC	0.89	0.97	0.90	0.90

(b) Dataset 2 - DT Model

Dataset	1	2	3	4
Accuracy	0.94	0.94	0.98	0.94
F1	0.84	0.84	0.96	0.83
Precision	0.84	0.83	0.95	0.83
Recall	0.84	0.85	0.96	0.84
ROC	0.90	0.90	0.97	0.90

(c) Dataset 3 - DT Model

Dataset	1	2	3	4
Accuracy	0.94	0.94	0.95	0.98
F1	0.85	0.85	0.85	0.96
Precision	0.85	0.85	0.85	0.96
Recall	0.85	0.84	0.85	0.96
ROC	0.91	0.90	0.91	0.97

(d) Dataset 4 - DT Model

Table 5.7: KNN model cross-performance results

Dataset	1	2	3	4
Accuracy	0.97	0.97	0.97	0.96
F1	0.93	0.91	0.90	0.90
Precision	0.96	0.95	0.94	0.94
Recall	0.90	0.87	0.87	0.87
ROC	0.94	0.93	0.93	0.93

(a) Dataset 1 - KNN Model

Dataset	1	2	3	4
Accuracy	0.96	0.97	0.96	0.96
F1	0.90	0.93	0.90	0.89
Precision	0.94	0.96	0.94	0.94
Recall	0.86	0.90	0.86	0.85
ROC	0.92	0.94	0.92	0.92

(b) Dataset 2 - KNN Model

Dataset	1	2	3	4
Accuracy	0.96	0.96	0.97	0.96
F1	0.90	0.90	0.93	0.90
Precision	0.95	0.95	0.96	0.94
Recall	0.86	0.86	0.89	0.86
ROC	0.92	0.92	0.94	0.92

(c) Dataset 3 - KNN Model

Dataset	1	2	3	4
Accuracy	0.96	0.96	0.96	0.97
F1	0.90	0.90	0.90	0.92
Precision	0.94	0.95	0.94	0.96
Recall	0.86	0.86	0.87	0.89
ROC	0.92	0.92	0.93	0.94

(d) Dataset 4 - KNN Model

Table 5.8: MLP model cross-performance results

Dataset	1	2	3	4
Accuracy	0.97	0.96	0.96	0.96
F1	0.92	0.89	0.89	0.89
Precision	0.97	0.95	0.94	0.95
Recall	0.88	0.85	0.85	0.85
ROC	0.93	0.92	0.92	0.92

(a) Dataset 1 - MLP Model

Dataset	1	2	3	4
Accuracy	0.96	0.97	0.96	0.96
F1	0.90	0.91	0.90	0.90
Precision	0.95	0.97	0.94	0.95
Recall	0.85	0.87	0.85	0.85
ROC	0.92	0.93	0.92	0.92

(b) Dataset 2 - MLP Model

Dataset	1	2	3	4
Accuracy	0.96	0.96	0.96	0.96
F1	0.89	0.89	0.90	0.88
Precision	0.89	0.89	0.90	0.89
Recall	0.88	0.88	0.91	0.88
ROC	0.93	0.93	0.94	0.93

(c) Dataset 3 - MLP Model

Dataset	1	2	3	4
Accuracy	0.95	0.95	0.95	0.95
F1	0.85	0.85	0.85	0.86
Precision	0.97	0.97	0.97	0.98
Recall	0.75	0.75	0.75	0.76
ROC	0.87	0.87	0.87	0.88

(d) Dataset 4 - MLP Model

Table 5.9: RF model cross-performance results

Dataset	1	2	3	4
Accuracy	0.99	0.97	0.97	0.97
F1	0.98	0.91	0.91	0.91
Precision	0.99	0.98	0.97	0.98
Recall	0.96	0.85	0.85	0.85
ROC	0.98	0.92	0.92	0.92

(a) Dataset 1 - RF Model

Dataset	1	2	3	4
Accuracy	0.96	0.99	0.97	0.97
F1	0.90	0.97	0.91	0.90
Precision	0.98	0.99	0.97	0.98
Recall	0.83	0.96	0.85	0.84
ROC	0.91	0.98	0.92	0.91

(b) Dataset 2 - RF Model

Dataset	1	2	3	4
Accuracy	0.96	0.97	0.99	0.97
F1	0.90	0.90	0.97	0.90
Precision	0.98	0.98	0.99	0.98
Recall	0.83	0.84	0.96	0.83
ROC	0.91	0.91	0.98	0.91

(c) Dataset 3 - RF Model

Dataset	1	2	3	4
Accuracy	0.96	0.97	0.97	0.99
F1	0.90	0.91	0.90	0.97
Precision	0.98	0.98	0.97	0.99
Recall	0.84	0.84	0.85	0.95
ROC	0.91	0.92	0.92	0.97

(d) Dataset 4 - RF Model

Table 5.10: SVM model cross-performance results

Dataset	1	2	3	4
Accuracy	0.97	0.97	0.97	0.97
F1	0.93	0.92	0.91	0.91
Precision	0.98	0.97	0.96	0.96
Recall	0.89	0.87	0.87	0.87
ROC	0.94	0.93	0.93	0.93

(a) Dataset 1 - SVM Model

Dataset	1	2	3	4
Accuracy	0.97	0.98	0.97	0.97
F1	0.91	0.94	0.91	0.91
Precision	0.97	0.98	0.96	0.96
Recall	0.86	0.89	0.86	0.86
ROC	0.92	0.94	0.93	0.92

(b) Dataset 2 - SVM Model

Dataset	1	2	3	4
Accuracy	0.97	0.97	0.98	0.97
F1	0.91	0.92	0.94	0.91
Precision	0.97	0.97	0.98	0.97
Recall	0.85	0.87	0.89	0.87
ROC	0.92	0.93	0.94	0.93

(c) Dataset 3 - SVM Model

Dataset	1	2	3	4
Accuracy	0.97	0.97	0.97	0.97
F1	0.91	0.91	0.91	0.93
Precision	0.97	0.96	0.96	0.98
Recall	0.85	0.86	0.86	0.88
ROC	0.92	0.93	0.93	0.94

(d) Dataset 4 - SVM Model

5.5 Testing Mathematical Model

To test the mathematical model we are using two models with single dataset. The said dataset contains 201 records of with total of 754 feature each row. The models used for the testing are random forest and support vector machine algorithm, which will be denoted as M_1 and M_2 respectively. These models and dataset is fed into the following tests cases.

CASE I: In this test case the we as user didnt provided any preference to the performance parameter. The system applied default weightage to the parameters.

CASE II: In this test case the user provided preference in this order: 1.Accuracy 2. Precision 3. Area Under ROC 4. F1 score 5. Recall. The time parameter was given slow preference ie careful approach.

CASE III: In this test case we used similar preference except for time parameter, where we opted for default preference.

CASE IV: Again in this test case we used similar preference except for time parameter, where we opted for fast preference.

Table 5.11: Weightage for test cases

Weights	Case I	Case II	Case III	Case IV
w_1	0.6	1.0	1.0	1.0
w_2	0.6	0.4	0.4	0.4
w_3	0.6	0.8	0.8	0.8
w_4	0.6	0.2	0.2	0.2
w_5	0.6	0.6	0.6	0.6
w_6	0.6	0.25	0.5	0.75

Table 5.12: Results of test cases

Case	Model 1's V_{score}	Model 2's V_{score}	Model selected
I	2.19907	2.26402	Model 2
II	2.29648	2.21802	Model 1
III	2.1574	2.19351	Model 2
IV	2.01834	2.16899	Model 2

The system used the mathematical model shown in eq. (3.1) to calculate the Vscores of the models. The system generated the weightage for the election process. This weightage is shown in table 5.11. Table 5.12 shows the Vscores of the models for test cases. As shown in the int table, Model 2 (Support Vector Machine model) was selected as the most-suited model for Case I, II, and IV. while Model 1 (the Random Forest model) was selected as the most-suited model for Case II. As we can see from table 5.12, the time parameter impacted the system quite significantly. Hence, we can assume that the implementation of the time factor is beneficial for the selection process.

Chapter 6

Applications

Chapter 6

Applications

The application is intended to be used by medical staff in COVID-19 wards. It is primarily built to detect arrhythmia in COVID-19 patients.

6.1 Detection of Arrhythmia

In the early phase of the COVID-19 pandemic, the mortality rate was extremely high. It was partly due to undetected comorbidities in patients. Undetected conditions were the primary cause of severe cases of COVID-19 infection. These conditions were respiratory disorders, cardiovascular disorders, diabetic disorders, etc. Earlier detection of these conditions will help in better case management. The project's application is to detect cardiovascular disorders. For this purpose, the presence of an arrhythmia can be used as a primary sign.

Arrhythmia can be detected from the patient's ECG reports. This process is automated with the application of a supervised learning system. This system allows medical staff to train and select the best-suited model for arrhythmia detection, and help them diagnose such patients. While the primary goal of this project is to be used in COVID-19 cases, it can also be used for regular patients.

6.2 Detection of anomalies in medical field

Machine learning is widely used for various tasks in the medical field. Supervised learning is used as diagnosing tool in a few cases. For example, epilepsy is diagnosed by studying scans obtained from the patient's brain. The proposed system can perform an initial screening to detect any anomalies. If such an anomaly is present, it will allow medical personnel to diagnose the patient efficiently.

The application of the system is not limited to diagnosis only, as it can be used for research purposes. These uses are filtering out unwanted data and detecting anomalies. This anomaly detection can be used in research data for discovery purposes. This will increase the productivity of research and lead to rapid development.

6.3 Model Training and Predictions

The system is capable of training models and selecting models. This system only needs a well-labeled dataset, and it will automatically select the best model for the task. The system can be used by a person without any technical or data science knowledge. The training process can be fine-tuned by adjusting performance weightage. The stored model can be used on similar datasets for prediction purposes.

Chapter 7

Conclusion And Future Scope

Chapter 7

Conclusion And Future Scope

7.1 Conclusion

As the number of COVID-19 are still increasing, the mortality rate is still rising. The mortality rate is higher in cases with comorbidities. Cardiovascular diseases are few of the comorbidities that drive this mortality rate. These diseases can be diagnosed earlier with a machine learning system. In this project we built an automated model training and selection system. This system can be used for predicting the presence of arrhythmia, which is the most common symptom in cardiovascular diseases.

The system uses supervised learning algorithms to generate models. This leads to efficient training and higher accuracy in predictions. The system can be fine tuned with by users, or it can be directly used without any formal training. The system showed good performance for similar datasets, so it can be used for general prediction purposes too.

7.2 Future Scope

The current system only provides solutions to binary classification problems, but it can be used for multiclass classification problems. The provision of user created model templates can be done, with user provided performance modification. The system can be connected directly to the hospital servers to train models from patient records directly with the help of the RPA system Ketkar and Gawade (2021).

References

Ağbulut, Ü., Gürel, A.E., Biçen, Y., 2021. Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison. Renewable and Sustainable Energy Reviews 135, 110114.

Alfaras, M., Soriano, M.C., Ortín, S., 2019. A fast machine learning model for ecg-based heartbeat classification and arrhythmia detection. Frontiers in Physics 7, 103.

Ayat, N.E., Cheriet, M., Suen, C.Y., 2005. Automatic model selection for the optimization of svm kernels. Pattern Recognition 38, 1733–1745.

Babapoor-Farrokhran, S., Rasekhi, R.T., Gill, D., Babapoor, S., Amanullah, A., 2020. Arrhythmia in covid-19. SN Comprehensive Clinical Medicine , 1–6.

Beri, A., Kotak, K., 2020. Cardiac injury, arrhythmia, and sudden death in a covid-19 patient. HeartRhythm case reports 6, 367–369.

Brazdil, P.B., Soares, C., Da Costa, J.P., 2003. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. Machine Learning 50, 251–277.

Burkart, N., Huber, M.F., 2021. A survey on the explainability of supervised machine learning. Journal of Artificial Intelligence Research 70, 245–317.

Chadaga, K., Prabhu, S., Vivekananda, B.K., Niranjana, S., Umakanth, S., 2021. Battling covid-19 using machine learning: A review. Cogent Engineering 8, 1958666.

Deo, R.C., 2015. Machine learning in medicine. Circulation 132, 1920–1930.

Fazeli, S., . ECG heartbeat categorization dataset. URL: `https://www.kaggle.com/shayanf azeli/heartbeat`. accessed on 15 Dec 2020.

Ghaderzadeh, M., Asadi, F., Hosseini, A., Bashash, D., Abolghasemi, H., Roshanpour, A., 2021. Machine learning in detection and classification of leukemia using smear blood images: a systematic review. Scientific Programming 2021.

Ghazal, T.M., Hasan, M.K., Alshurideh, M.T., Alzoubi, H.M., Ahmad, M., Akbar, S.S., Al Kurdi, B., Akour, I.A., 2021. Iot for smart cities: Machine learning approaches in smart healthcare—a review. Future Internet 13, 218.

Hong, S., Xiao, C., Ma, T., Li, H., Sun, J., 2019. Mina: multilevel knowledge-guided attention for modeling electrocardiography signals. arXiv preprint arXiv:1905.11333 .

Ibrahim, I., Abdulazeez, A., 2021. The role of machine learning algorithms for diagnosing diseases. Journal of Applied Science and Technology Trends 2, 10–19.

Ketkar, Y., Gawade, S., 2021. Effectiveness of robotic process automation for data mining using uipath, in: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), pp. 864–867. doi:10.1109/ICAIS50930.2021.9396024.

Kim, I.K., Lee, K., Park, J.H., Baek, J., Lee, W.K., 2021. Classification of pachychoroid disease on ultrawide-field indocyanine green angiography using auto-machine learning platform. British Journal of Ophthalmology 105, 856–861.

Lee, J.H., Lin, C.J., 2000. Automatic model selection for support vector machines. CSIE, NTU .

Liu, Q., Chen, H., Zeng, Q., 2020. Clinical characteristics of covid-19 patients with complication of cardiac arrhythmia. The Journal of Infection 81, e6.

Maschler, B., Weyrich, M., 2021. Deep transfer learning for industrial automation: a review and discussion of new techniques for data-driven machine learning. IEEE Industrial Electronics Magazine 15, 65–75.

Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A., Foozy, C.F.M., 2021. Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset. IEEE access 9, 22351–22370.

Mulia, E.P.B., Maghfirah, I., Rachmi, D.A., Julario, R., 2021. Atrial arrhythmia and its association with covid-19 outcome: a pooled analysis. Diagnosis .

Pande, S., Khamparia, A., Gupta, D., Thanh, D.N., 2021. Ddos detection using machine learning technique, in: Recent Studies on Computational Intelligence. Springer, pp. 59–68.

Ren, H.G., Guo, X., Tu, L., Hu, Q., Blighe, K., Safdar, L.B., Stebbing, J., Weiner, S.D., Willis, M.S., Rosendaal, F.R., et al., 2020. Clinical characteristics and risk factors for myocardial injury and arrhythmia in covid-19 patients. medRxiv .

Soman, T., Bobbie, P.O., 2005. Classification of arrhythmia using machine learning techniques. WSEAS Transactions on computers 4, 548–552.

Yarmohammadi, H., Morrow, J.P., Dizon, J., Biviano, A., Ehlert, F., Saluja, D., Waase, M., Elias, P., Poterucha, T.J., Berman, J., et al., 2021. Frequency of atrial arrhythmia in hospitalized patients with covid-19. The American Journal of Cardiology 147, 52–57.

Zhang, D., Shen, Y., Huang, Z., Xie, X., 2022. Auto machine learning-based modelling and prediction of excavation-induced tunnel displacement. Journal of Rock Mechanics and Geotechnical Engineering .

Zuranski, A.M., Martinez Alvarado, J.I., Shields, B.J., Doyle, A.G., 2021. Predicting reaction yields via supervised learning. Accounts of chemical research 54, 1856–1865.

List of Publications

Y. Ketkar and S. Gawade, **"Effectiveness of Robotic Process Automation for data mining using UiPath"**, in *2021 International Conference on Artificial Intelligence and Smart Systems(ICAIS)*. IEEE, 2021,pp. 864–867.

Y. Ketkar and S. Gawade, **"Novel approach to identify suitable machine learning model in the healthcare industry with user prefered parameters."**, Under Revision *2022, Healthcare Analytics*.

Y. Ketkar and S. Gawade, **"Detection of Arrhythmia Using Weightage-based Supervised Learning System for COVID-19."**, Under Revision *2022, Intelligent Systems with Applications*.

ACKNOWLEDGEMENT

This dissertation report would not have come into reality without the able guidance, support and wishes of all those who stand by me in the development. I wish to give my special thanks to my guide, **Dr. Sushopti Gawade**, for his/her timely advice and guidance.

I would like to thank our Principal, **Dr. Sandeep Joshi** for his constant encouragement throughout the course. I humbly thank our M.E Coordinator, **Dr. Prashant Nitnaware** and our Head of Department, **Dr. Satishkumar Verma**, for their valuable guidance & unending support despite a very busy work schedule. The cheerful spirit they radiated all the time fuelled our desire to excel in the work that I had undertaken.

I acknowledge all the staff members of the department of Computer Engineering and department of Information Technology for their help and suggestions during various phases of this project work. It's difficult to forget my eminent supporters that are my Friends and Family members who are always there encouraging me in my every deed.

Yashodhan Prakash Ketkar