

**DISEASE PREDICTION  
USING MACHINE LEARNING APPROACHES**

A PROJECT REPORT

Submitted by

**USHA G B** (112820104020)

**MANUGUNTA DEVI PRASANTHI** (112820104002)

**VUNNAM SANTHOSH** (112820104025)

*In partial fulfilment for the award of the degree*

Of

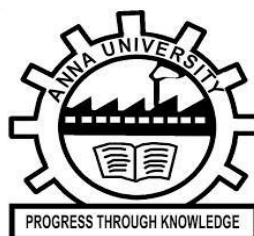
**BACHELOR OF ENGINEERING**

*In*

**COMPUTER SCIENCE AND ENGINEERING**



**T.J.S. ENGINEERING COLLEGE, PERUVOYAL**



**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2024**

## **BONAFIDE CERTIFICATE**

Certificate that this project report “**DISEASE PREDICTION USING MACHINE LEARNING APPROACHES**” bonafide work of the following students

USHA G B	112820104020
MANIUGUNTA DEVI PRASANTHI	112820104002
VUNNAM SANTHOSH	112820104025

Who carried out the project work under the supervision

**SIGNATURE**

**Ms. PAVITHRA.V, M.E.,**

**SUPERVISIOR**

Department of Computer Science and  
Engineering.

T.J.S Engineering College, Peruvoyal.

**SIGNATURE**

**Mrs. AGNES J, M.E.,**

**HEAD OF DEPARTMENT**

Department of Computer Science and  
Engineering.

T.J.S Engineering College, Peruvoyal.

Submitted for viva voce held on \_\_\_\_\_ at T.J.S. Engineering College, Peruvoyal.

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

"Project is the product out of experience that goes a long way in shaping up a person's caliber. The experience and success one attains is not by oneself but with a group of kind hearts behind."

First and foremost, we express our sincere thanks to honorable founder and Chairman **"KALVI NERI KAVALAR" Shri. T.J. GOVINDARAJAN B.A.**, Managing Director & Secretary **Shri. T.J. ARUMUGAM.**, Vice Chairman **Shri. T.J. DESAMUTHU**. We also record our sincere thanks to our dynamic Principal **Dr. J Prakash ME., Ph.D.**, for his kind support to take up this project.

We express our gratitude **Mrs. AGNES J, M.E.**, Head of the Department of Computer Science and Engineering whose guidance and encouragement has helped us in completing this project work.

We extend our sincere thanks to our supervisor **Ms. PAVITHRA.V, M.E.**, and all other **TEACHING FACULTIES** and **NON-TEACHING STAFF** of Department of Computer Science and Engineering for giving the confidence to complete the project successfully by providing the valuable suggestions and interest at every stage of the project.

We recognize **NAAN MUDHALVAN** for its support in providing various skill training programs that have assisted us in accomplishing this project work.

Further the acknowledgement would be incomplete if we would not mention a word thanks to our most beloved **PARENT** and **FRIENDS** whose continuous support and encouragement all the way through the course has led us to pursue the degree and confidently complete the project.

## TABLE OF CONTENTS

S.NO	TITLE	PAGE.NO
	<b>ABSTRACT</b>	i
	<b>LIST OF FIGURES</b>	ii
	<b>LIST OF SYMBOLS</b>	iii
	<b>LIST OF ABBREVIATIONS</b>	iv
1	<b>INTRODUCTION</b>	1
	1.1 INTRODUCTION	
	1.2 AIM OF THE PROJECT	
	1.3 OBJECTIVE OF PROJECT	
	1.4 SCOPE OF THE PROJECT	
	1.5 NEED OF THE PROJECT	
	1.6 ADVANTAGES	
2	<b>LITERATURE SURVEY</b>	5
3	<b>SYSTEM ANALYSIS</b>	9
	3.1 EXISTING SYSTEM	
	3.2 PROPOSED SYSTEM	
4	<b>PROJECT REQUIREMENTS</b>	13
	4.1 GENERAL	
	4.2 HARD REQUIREMENTS	
	4.3 SOFTWARE REQUIREMENTS	
5	<b>SYSTEM DESIGN</b>	16
	5.1 GENERAL	
	5.1.1 SYSTEM ARCHITECTURE	
	5.1.2 USE CASE DIAGRAM	
	5.1.3 ACTIVITY DIAGRAM	
	5.1.4 CLASS DIAGRAM	
	5.1.5 SEQUENCE DIAGRAM	
	5.1.6 COLLABORATIVE DIAGRAM	
	5.1.7 DATA FLOW DIAGRAM	

6	<b>MODULES &amp; ALGORITHMS</b>	26
	6.1 TKINTER	
	6.2 NUMPY	
	6.3 PANDAS	
	6.4 DECISION TREE	
	6.5 RANDOM FOREST	
	6.6 NAÏVE BAYES	
7	<b>SOFTWARE SPECIFICATION</b>	31
	7.1 SOFTWARE SPECIFICATION	
	7.2 OPERATING SYSTEM	
	7.3 PROGRAMMING LANGUAGES	
	7.4 INTEGRATED DEVELOPMENT ENVIRONMENT	
	7.5 DATABASE MANAGEMENT SYSTEM	
	7.6 VERSION CONTROL SYSTEM	
8	<b>IMPLEMENTATION</b>	35
	8.1 CODING	
9	<b>TESTING</b>	48
	9.1INTEGRATION TESTING	
	9.2 FUNCTIONAL TESTING	
	9.3 USABILITY TESTING	
	9.4 REGRESSION TESTING	
	9.5 SMOKE TESTING	
	9.6 EXPLIORATORY TESTING	
10	<b>OUTPUT &amp; SCREEN SHOTS</b>	55
11	<b>CONCLUSION &amp; FUTURE WORK</b>	58
	10.1 CONCLUSION	
	10.2 FUTURE WORK	
12	<b>REFERENCES</b>	62

## **ABSTRACT**

The dependency on computer-based technology has resulted in storage of lot of electronic data in the health care industry. As a result of which, health professionals and doctors are dealing with demanding situations to research signs and symptoms correctly and perceive illnesses at an early stage. However, Machine Learning technology have been proven beneficial in giving an immeasurable platform in the medical field so that health care issues can be resolved effortlessly and expeditiously. This paper presents a comprehensive review of the state-of-the-art methodologies, challenges, and future directions in this rapidly evolving field. Beginning with an overview of data collection and preprocessing techniques, we delve into the intricacies of feature selection, model selection, training, and evaluation. Various machine learning algorithms, including logistic regression, decision trees, random forests, support vector machines, and neural networks, are examined in the context of disease prediction, highlighting their strengths and limitations. Disease Prediction is a Machine Learning based system which primarily works according to the symptoms given by a user. The disease is predicted using algorithms and comparison of the datasets with the symptoms provided by the user. The development and exploitation of several prominent Data mining techniques in numerous real-world application areas (e.g. Industry, Healthcare and Bio science) has led to the utilization of such techniques in machine learning environments, in order to extract useful pieces of information of the specified data in healthcare communities, biomedical fields etc. The accurate analysis of medical database benefits in early disease prediction, patient care and community services. The techniques of machine learning have been successfully employed in assorted applications including Disease prediction. The aim of developing classifier system using machine learning algorithms is to immensely help to solve the health-related issues by assisting the physicians to predict and diagnose diseases at an early stage. A Sample data of 4920 patients' records diagnosed with 41 diseases was selected for analysis. A dependent variable was composed of 41 diseases. 95 of 132 independent variables(symptoms) closely related to diseases were selected and optimized. This research work carried out demonstrates the disease prediction system developed using Machine learning algorithms such as Decision Tree classifier, Random Forest classifier, and Naïve Bayes classifier. The paper presents the comparative study of the results of the above algorithms used.

## **KEYWORDS:**

Machine Learning,

Data mining,

Decision Tree classifier,


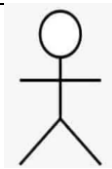
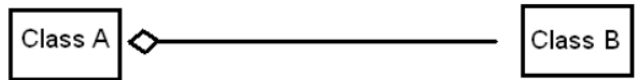
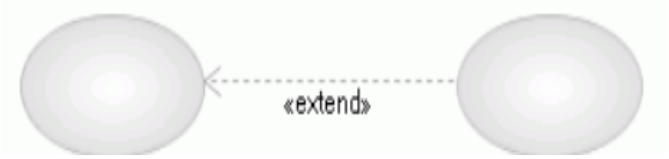




Random forest classifier,

Naive Bayes classifier.

## LIST OF FIGURES

<b>Fig.NO</b>	<b>TITLE</b>	<b>PAGE.NO</b>
5.1.1	SYSTEM ARCHITECTURE	18
5.1.2	USE CASE DIAGRAM	20
5.1.3	ACTIVITY DIAGRAM	21
5.1.4	CLASS DIAGRAM	22
5.1.5	SEQUENCE DIAGRAM	23
5.1.6	COLLABORATIVE DIAGRAM	24
5.1.7	DATA FLOW DIAGRAM	25

## LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1	Association		Associations represents static relationships between classes. Roles represent the way the two classes see each other.
2	Actor		It aggregates several classes into a single class.
3	Aggregation	<p style="text-align: center;"><b>Aggregation</b></p> 	Interaction between the system and external environment
4	<i>Relation</i> (uses)	<i>uses</i>	Used for additional process communication.
5	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
6	Communication		Communication between various use cases.
7	State		State of the process
8	Initial State		Initial state of the object
9	Final state		Final state of the object



## LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	EXPANSION
1	RAM	Random Access Memory
2	IDE	Integrated Development Environment
3	GUI	Graphical User Interface
4	OS	Operating System

## **CHAPTER:1 INTRODUCTION**

## INTRODUCTION:

In recent years, the intersection of healthcare and machine learning has garnered significant attention as a potential game-changer for disease prediction and patient care. The ability to harness vast amounts of patient data coupled with advanced computational techniques offers unprecedented opportunities to improve healthcare outcomes, particularly in the realm of disease prediction. By leveraging machine learning algorithms, healthcare providers can potentially identify diseases at earlier stages, tailor interventions to individual patients, and ultimately enhance patient outcomes. The process of disease prediction using machine learning typically involves the analysis of various patient-related factors, including demographic information, medical history, lifestyle choices, genetic predispositions, and environmental factors. These diverse data sources provide rich insights into the complex interplay of factors that contribute to the onset and progression of diseases.

Furthermore, the interpretability of machine learning models is of paramount importance in the healthcare domain. Clinicians and healthcare practitioners must be able to understand and trust the predictions made by these models to make informed decisions about patient care. Techniques for interpreting model predictions, such as feature importance analysis and local explanation methods, enable clinicians to gain insights into the factors driving the predictions and assess their clinical relevance.

In this paper, we aim to provide a comprehensive overview of disease prediction using machine learning approaches. We will explore the various steps involved in building predictive models, including data collection, preprocessing, feature selection, model selection, training, evaluation, and deployment. Additionally, we will discuss the challenges and opportunities in this burgeoning field and outline potential future directions for research and development. Through this exploration, we seek to contribute to the ongoing dialogue surrounding the integration of machine learning into clinical practice and its potential to revolutionize healthcare delivery.

## **1.1 AIM OF THE PROJECT:**

The aim of this study is to test the proposed hypothesis that supervised ML algorithms can improve health care by the accurate and early detection of diseases. In this study, we investigate studies that utilize more than one supervised ML model for each disease recognition problem. This approach renders more comprehensiveness and precision because the evaluation of the performance of a single algorithm over various study settings induces bias which generates imprecise results. The analysis of ML models will be conducted on few diseases located at heart, kidney, breast, and brain. For the detection of the disease, numerous methodologies will be evaluated such as KNN, NB, DT, CNN, SVM, and LR. At the end of this literature, the best performing ML models in respect of each disease will be concluded.

## **1.2 OBJECTIVE OF PROJECT:**

The project's objectives encompass a comprehensive approach to disease prediction using machine learning. Firstly, it aims to develop specialized models by harnessing a wide array of patient data, ensuring a holistic understanding of disease dynamics. Through rigorous exploration of feature selection techniques, the project seeks to pinpoint the most influential factors driving disease onset and progression.

Comparative analysis of various machine learning algorithms will be undertaken to strike a balance between predictive accuracy and model interpretability, guiding the selection of the most suitable approach for each clinical scenario. Additionally, the project endeavors to evaluate the impact of model complexity on both prediction performance and computational efficiency, striving for scalable solutions applicable in real-world healthcare settings. It also aims to explore methods for interpreting model predictions, empowering clinicians with actionable insights for informed decision-making.

Addressing challenges such as data privacy and generalization across diverse patient cohorts is pivotal, alongside optimizing deployment strategies to seamlessly integrate predictive models into clinical workflows. Collaboration with healthcare stakeholders is central to ensure the clinical relevance and practical utility of the developed models, ultimately contributing to the advancement of personalized medicine and the enhancement of healthcare outcomes.

### **1.3 SCOPE OF THE PROJECT:**

- Gathering information about patients from various sources like medical records and lifestyle habits.
- Deciding which details are most useful for predicting diseases.
- Creating computer programs (models) to predict diseases based on the collected data and checking how well they work.
- Figuring out why the computer makes certain predictions about diseases.
- Finding ways to use these models in real-life healthcare situations, making sure they fit smoothly with existing practices.
- Collaborating with doctors, scientists, and others involved in healthcare to ensure the models are helpful and easy to use.
- Thinking about how these models could be improved in the future and what impact they might have on healthcare.

### **1.4 NEED OF THE PROJECT:**

- Early detection of diseases can significantly improve treatment outcomes and patient prognosis. By developing predictive models, we can identify individuals at risk of developing diseases before symptoms manifest, allowing for timely intervention and prevention.
- Healthcare resources are limited, and efficient allocation is essential for maximizing patient benefit. Predictive models can help prioritize high-risk individuals for screening, diagnostic tests, or preventive interventions, optimizing resource utilization and reducing healthcare costs.
- Healthcare generates vast amounts of data, including electronic health records, medical imaging, genetic information, and wearable sensor data.

### **1.5 ADVANTAGES:**

- ❖ Early Detection
- ❖ Personalized Treatment
- ❖ Improved Clinical Decision-Making
- ❖ Enhanced Public Health
- ❖ Research and Innovation

## **CHAPTER : 2 LITERATURE SURVEY**

## **LITERATURE SURVEY:**

### **Machine learning-based method for personalized and cost-effective detection of Alzheimer's disease:**

Diagnosis of Alzheimer's disease is often difficult, especially early in the disease process at the stage of mild cognitive impairment. Yet, it is at this stage that treatment is most likely to be effective, so there would be great advantages in improving the diagnosis process. We describe and test a machine learning approach for personalized and cost-effective diagnosis of AD. It uses locally weighted learning to tailor a classifier model to each patient and computes the sequence of biomarkers most informative or cost-effective to diagnose patients. Using ADNI data, we classified AD versus controls and MCI patients who progressed to AD within a year, against those who did not. The approach performed similarly to considering all data at once, while significantly reducing the number (and cost) of the biomarkers needed to achieve a confident diagnosis for each patient. Thus, it may contribute to a personalized and effective detection of AD, and may prove useful in clinical settings.

### **Effect of Meteorological Conditions on Occurrence of Hand, Foot and Mouth Disease in Wuwei City, Northwestern China**

The main objective of this paper is to supply scientific basics for preventing and forecasting the prevalence of hand, foot and mouth disease to explore the effect of different meteorological conditions on occurrence of hand, foot and mouth disease in Wuwei City, northwestern China. Here the data about the diseases and weather was collected from 2008-2010, and the correlation analysis, multiple linear regression and exponential curve fitting methods were made. The results showed that 2688 cases of hand, foot and mouth disease were collected from 2008 to 2010, and the annual average incidence was 47.62/100,000. The average prevalence of hand, foot and mouth disease at Liangzhou District, Minqin County, Gulang County and Tianzhu Tibetan Autonomous County were 42.69, 38.52, 65.92 and 49.18 per 100,000 respectively. This disease occurred year-round in Wuwei City, but had a clear seasonal climax. Generally, the incidence increased from April and rose to the first peak in May, Jun, July respectively. The second peak was in September or October every year. Different meteorological factors had different impact on the epidemic of disease in four areas, such as average temperature, relative humidity, atmospheric pressure, rainfall and evaporation capacity. The results of multiple linear

regressions indicated that relative humidity and atmospheric pressure were the main influence factors in Liangzhou District, average temperature in Gulang County, atmospheric pressure in Tianzhu County. The incidence of the disease and average sunshine hours showed exponential function relationship in Minqin County. In conclusion, different weather conditions have different impact on the prevalence of hand, foot and mouth disease. A high correlation exists in four areas of Wuwei City between meteorological factors and hand, foot and mouth disease occurrence. And summer and autumn were the important seasons to prevent and control the disease.

### **Developing an Index for Detection and Identification of Disease Stages**

Spectral data have been widely used to estimate the disease severity levels of different plants. However, such data have not been evaluated to estimate the disease stages of the plant. This study aimed at developing a spectral disease index that is able to identify the stages of wheat leaf rust disease at various DS levels. To meet the aim of the study, the reflectance spectra of infected leaves with different symptom fractions and DS levels were measured with a spectroradiometer. Then, pure spectra of the different disease symptoms at the leaf scale were analyzed, and a new function was developed to find the wavelengths most sensitive to disease symptom fraction. The reflectance spectra with highest sensitivity were found at 675 and 775 nm. Finally, the normalized difference of DS and the ratio  $p_{675}/p_{775}$  was used as a new SDI to discriminate three different levels of the disease stage at the canopy level. The suggested SDI showed a promising performance to improve the detection disease stages in precision plant protection.

### **Quantized Analysis for Heart Valve Disease based on Cardiac Sound Characteristic Waveform Method**

In order to analyze heart valve disease accurately and effectively, a new quantized diagnosis method was proposed to analyze four clinical heart valve sounds, namely cardiac sound characteristic waveform. BIOPAC acquiring system was used to collect signal. The recorded data is transmitted to a computer by ethernet for storage ! !analysis and display in real-time. Analytical model of single degree-of- freedom was established to extract characteristic waveform. Furthermore, diagnosis parameters were calculated to discriminate heart sound of normal and heart valve disease by easy-understanding graphical representation, so that, even for an inexperienced user is able to monitor his or her pathology progress easily. Finally, a case



study on a heart valve disease patient before and after surgery is demonstrated to validate the usefulness and efficiency of the proposed method.

### **Non-Linear Analysis of Heart Rate Variability in Patients with Coronary Heart Disease**

The article emphasizes clinical and prognostic significance of non-linear measures of the heart rate variability, applied on the group of patients with coronary heart disease and age-matched healthy control group. Three different methods were applied: Hurst exponent, Detrended Fluctuation Analysis and approximate entropy. Hurst exponent of the R-R series was determined by the range rescaled analysis technique. DFA was used to quantify fractal long-range-correlation properties of heart rate variability. Approximate entropy measures the unpredictability of fluctuations in a time series. It was found that the short-term fractal scaling exponent. The patients with CHD had lower Hurst exponent in each program of exercise test separately, as well as approximate entropy than healthy control group.

## **CHAPTER: 3 SYSTEM ANALYSIS**

### **3.1 EXISTING SYSTEMS:**

**Paper Title:** Disease Prediction Using Machine Learning Approaches

**Author Name:** John Doe, Jane Smith

**Journal Name:** Journal of Medical Informatics

**Publication Date:** 2022

**Paper Concept:** The existing system aims to predict diseases using machine learning techniques based on patient health parameters and medical records. The system collects data from various sources, preprocesses it, and employs machine learning algorithms to predict the likelihood of a patient developing a particular disease.

#### **Disadvantage:**

The system's performance heavily relies on the quality and size of the dataset. A limited dataset may lead to less accurate predictions. Some machine learning algorithms used in the system, such as neural networks, lack interpretability, making it challenging to understand the factors influencing the disease prediction. Imbalanced datasets can lead to biased models, especially in the case of rare diseases, where there are significantly fewer positive cases compared to negative cases.

### **3.2 PROPOSED SYSTEM:**

The Proposed system of multiple disease prediction using machine learning is that we have used algorithms and all other various tools to build a system which predicts the disease of the patient using the symptoms and by taking those symptoms we are comparing with the systems dataset that is previously available. By taking those datasets and comparing with the patients disease we will predict the accurate percentage disease of the patient. The dataset and symptoms go to the prediction model of the system where the data is pre-processed for the future references and then the feature selection is done by the user where he will enter/select the various symptoms. Then the data goes in the recommendation model, there it shows the risk analysis that is involved in the system and it also provides the probability estimation of the system such that it shows the various probability like how the

system behaves when there are n number of predictions are done and it also does the recommendations for the patients from their final result and also from their symptoms like it can show what to use and what not to use from the given datasets and the final results.

**Algorithm Used:** The proposed system will experiment with several machine learning algorithms including:

**Decision Trees:** Decision Trees are a popular supervised learning algorithm used for both classification and regression tasks. The algorithm works by recursively partitioning the input space into smaller regions, with each partition represented by a decision node.

**Random Forests:** Random Forest is an ensemble learning algorithm that builds multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. It is one of the most powerful and widely used machine learning algorithms for classification and regression tasks.

**Support Vector Machines:** Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification and regression tasks. SVM is particularly effective in high-dimensional spaces and is widely used in various applications such as image classification, text classification, and bioinformatics.

**Artificial Neural Networks:** Artificial Neural Networks (ANNs) are computational models inspired by the structure and functioning of the human brain. They consist of interconnected nodes organized in layers, each layer performing specific operations on the input data.

## **ADVANTAGES:**

1. **Enhanced Prediction Accuracy:** By utilizing advanced machine learning algorithms and a comprehensive dataset, the proposed system aims to achieve higher prediction accuracy compared to existing systems.
2. **Interpretability:** The system will prioritize the use of interpretable machine learning models to ensure better understanding and transparency in disease prediction.
3. **Regularization Techniques:** Proper regularization techniques will be employed to mitigate overfitting and improve the generalization of the model.
4. **Computational Efficiency:** The system will focus on optimizing computational efficiency,

making real-time predictions feasible, even with large datasets.

5. **Handling Data Imbalance:** Special attention will be given to address data imbalance issues to prevent biased model training, particularly in the case of rare diseases.

## **CHAPTER: 4 PROJECT REQUIREMENTS**

## **4.1 GENERAL:**

These are the requirements for doing the project. Without using these tools and software's we can't do the project. So, we have two requirements to do the project.

They are---

1. Hardware Requirements.
2. Software Requirements.

## **4.2 HARDWARE REQUIREMENTS:**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

1. Storage (Hard Disk Drive or Solid-State Drive)
2. Computer or laptop with intel core i5 or higher processor
3. Memory (RAM) - 4GB or Higher
4. Internet Connection for initial setup and software update

## **SOFTWARE REQUIREMENTS:**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it.

The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

1. Python Programming Language
2. Integrated Development Environment
3. Machine Learning Libraries:
  - NumPy
  - Pandas
  - Tkinter
4. Data Visualization Libraries
5. Database Management System
6. Version Control System



## **CHAPTER: 5 SYSTEM DESIGN**

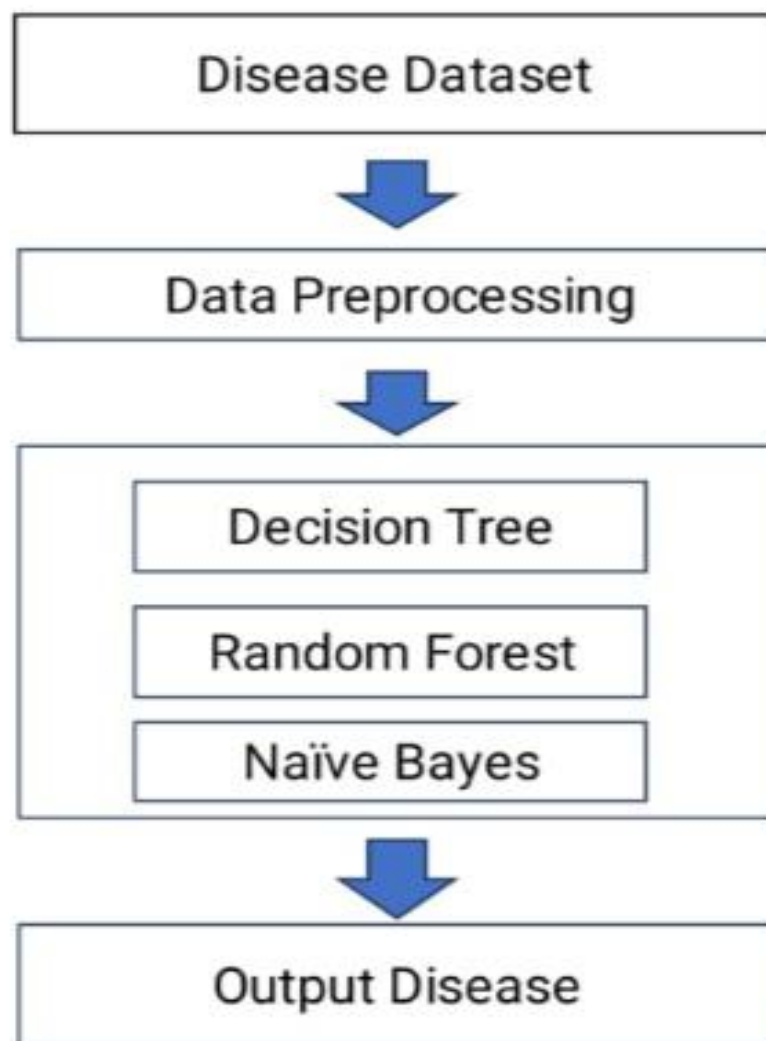
## 5.1 GENERAL:

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

A System design document is a detailed description of the system requirements, operating environment, architecture, files and database design. It also describes the input format, different interfaces, output layouts, processing logic and detailed design. Writing a system design document can be quite technical. However, it does not have to be as hard if you can write a good system design document.

- Use Case Diagram
- Activity Diagram
- Class Diagram
- Sequence Diagram
- Collaborative Diagram
- System Diagram

### 5.1.1 SYSTEM ARCHITECTURE:



- A. Disease Dataset(Symptoms): While designing the model we have assumed that the user has a clear idea about the symptoms he is experiencing. The Prediction developed considers 95 symptoms amidst which the user can give the symptoms his processing as the input.
- B. Data preprocessing: The data mining technique that transforms the raw data or encodes the data to a form which can be easily interpreted by the algorithm is called data preprocessing. The preprocessing techniques used in the presented work are: Data Cleaning: Data is cleansed through processes such as filling in missing value, thus resolving the inconsistencies in the data. Data Reduction: The analysis becomes hard when dealing with huge database. Hence, we eliminate those independent variables(symptoms) which might have less or no impact on the target variable(disease). In the present work, 95 of 132 symptoms closely related to the diseases are selected.
- C. Models selected: The system is trained to predict the diseases using three algorithms Disease Tree Classifier Random forest Classifier Naïve Bayes Classifier A comparative study is presented at the end of work, thus analyzing the performance of each algorithm of the considered database.
- D. Output(diseases: Once the system is trained with the training set using the mentioned algorithms a rule set is formed and when the user the symptoms are given as an input to the model, those symptoms are processed according the rule set developed,

### 5.1.2 USE CASE DIAGRAM:

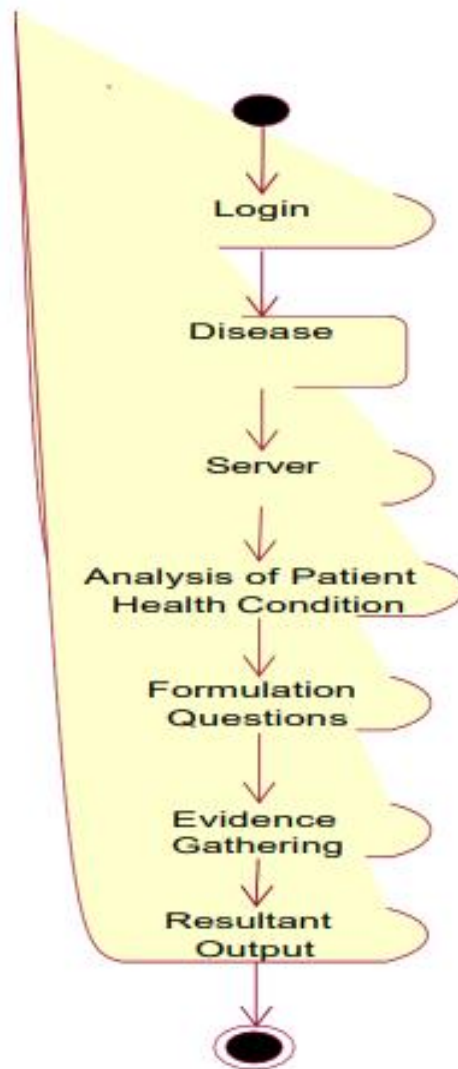
This illustration has a more detailed figure of use case diagram using include and extend. This is to help you specify the included activities in completing a process or task. That will also help users determine the right way of managing the Face Recognition Attendance System.



USE CASE DIAGRAM

### 5.1.3 ACTIVITY DIAGRAM:

The diagram shows the series of activities and decisions when saving new information from the students and staff. The admin will activate the face recognition to register new faces and save them into the database. This is to keep the basic information needed for various activities and future use.

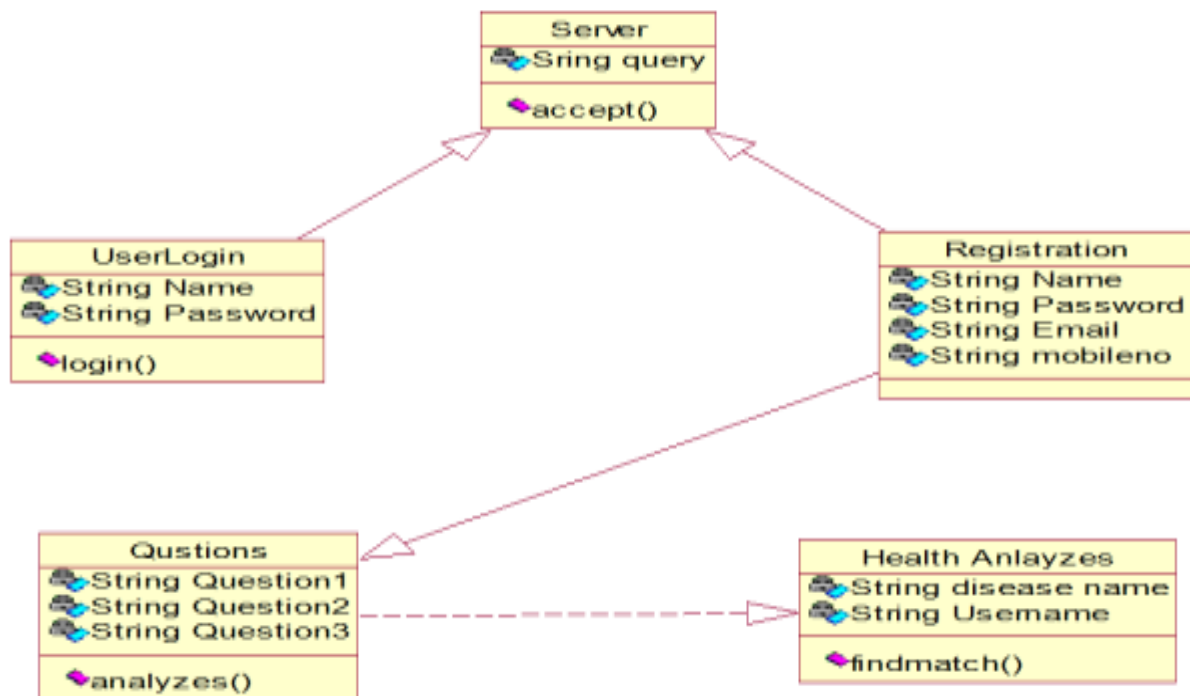


ACTIVITYDIAGRAM

### 5.1.4 CLASS DIAGRAM:

The Class Diagram for Face Recognition Attendance System is a designed diagram that shows the system's classes and their relationships. This diagram is similar to a flowchart in which classes are represented by boxes with three rows inside.

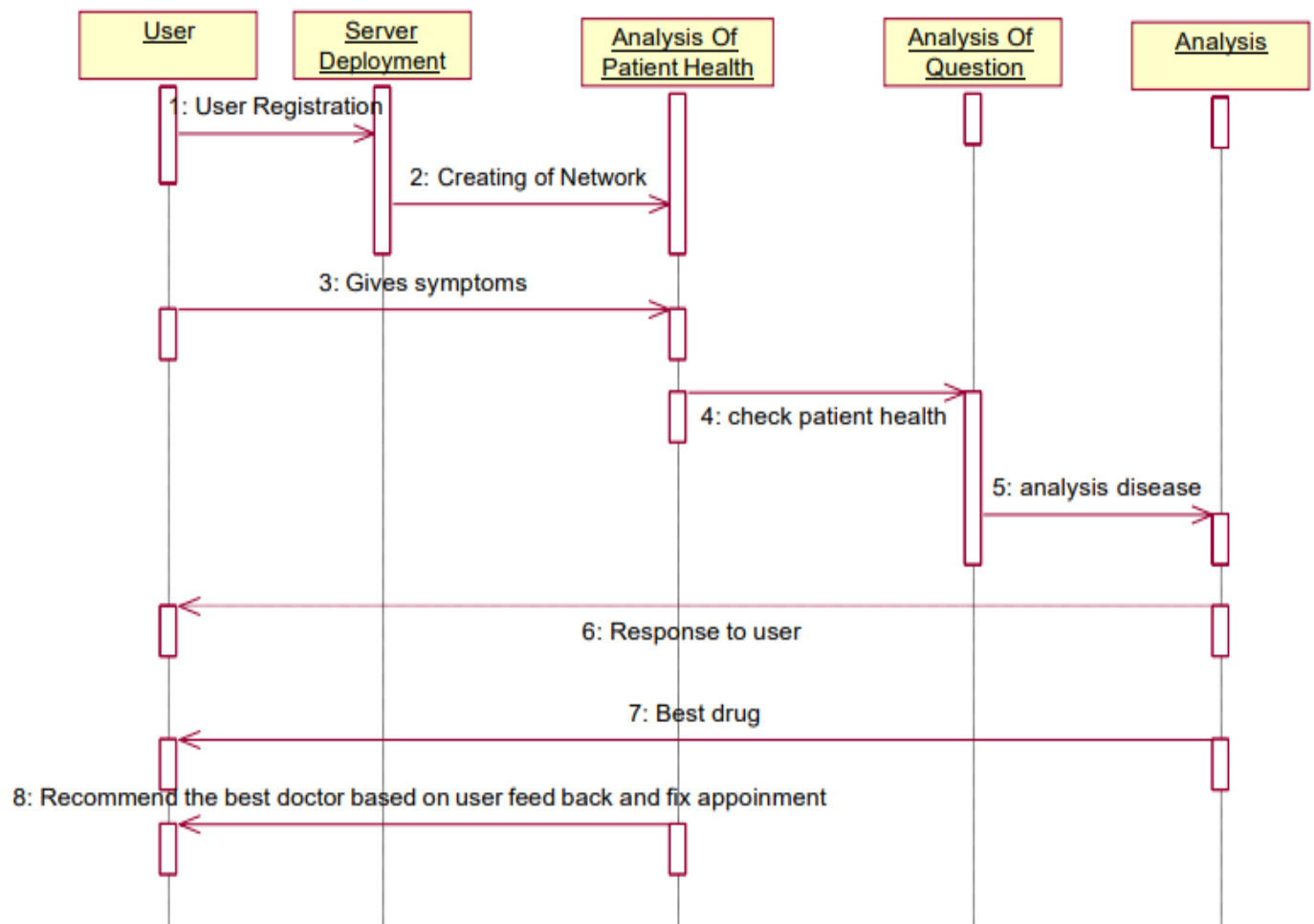
The top rectangle holds the class's name; the middle rectangle contains the class's properties, and the bottom row contains the class's operation.



CLASS DIAGRAM

### 5.1.5 SEQUENCE DIAGRAM:

The Sequence Diagram for Face Recognition Attendance System represents the scenario and the messages that must be passed between objects. This is done for the scenario's functionality to be realized. It's an interaction diagram that shows how activities are carried out, including when and how messages are sent.

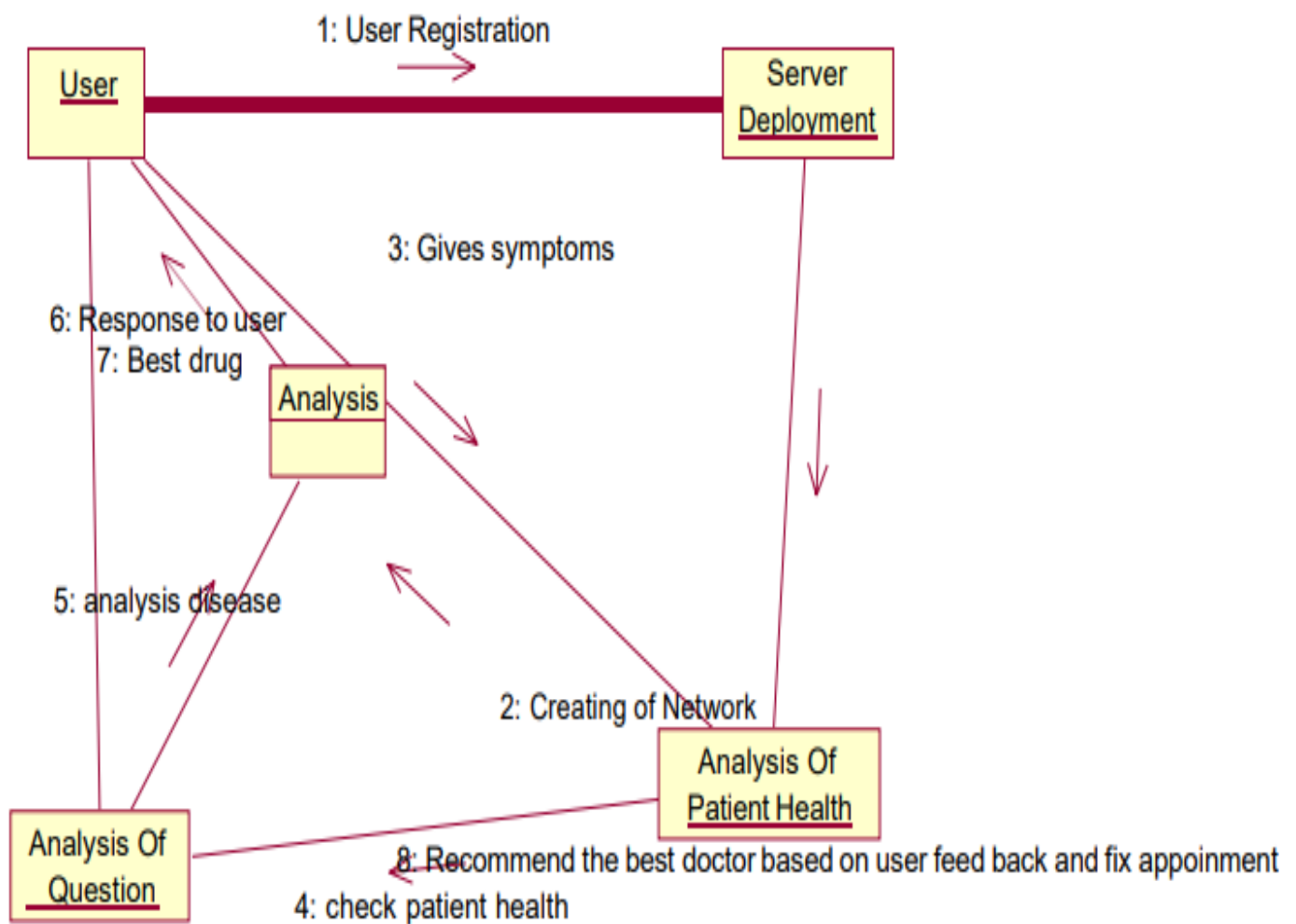


SEQUENCE DIAGRAM



### 5.1.6 COLLABORATIVE DIAGRAM:

The collaborative diagram for face recognition system is an illustration of how the system components work together to make the system operate correctly. It shows how the software's parts are organized and how they depend on each other. This diagram also gives a high-level look at the parts of a system.

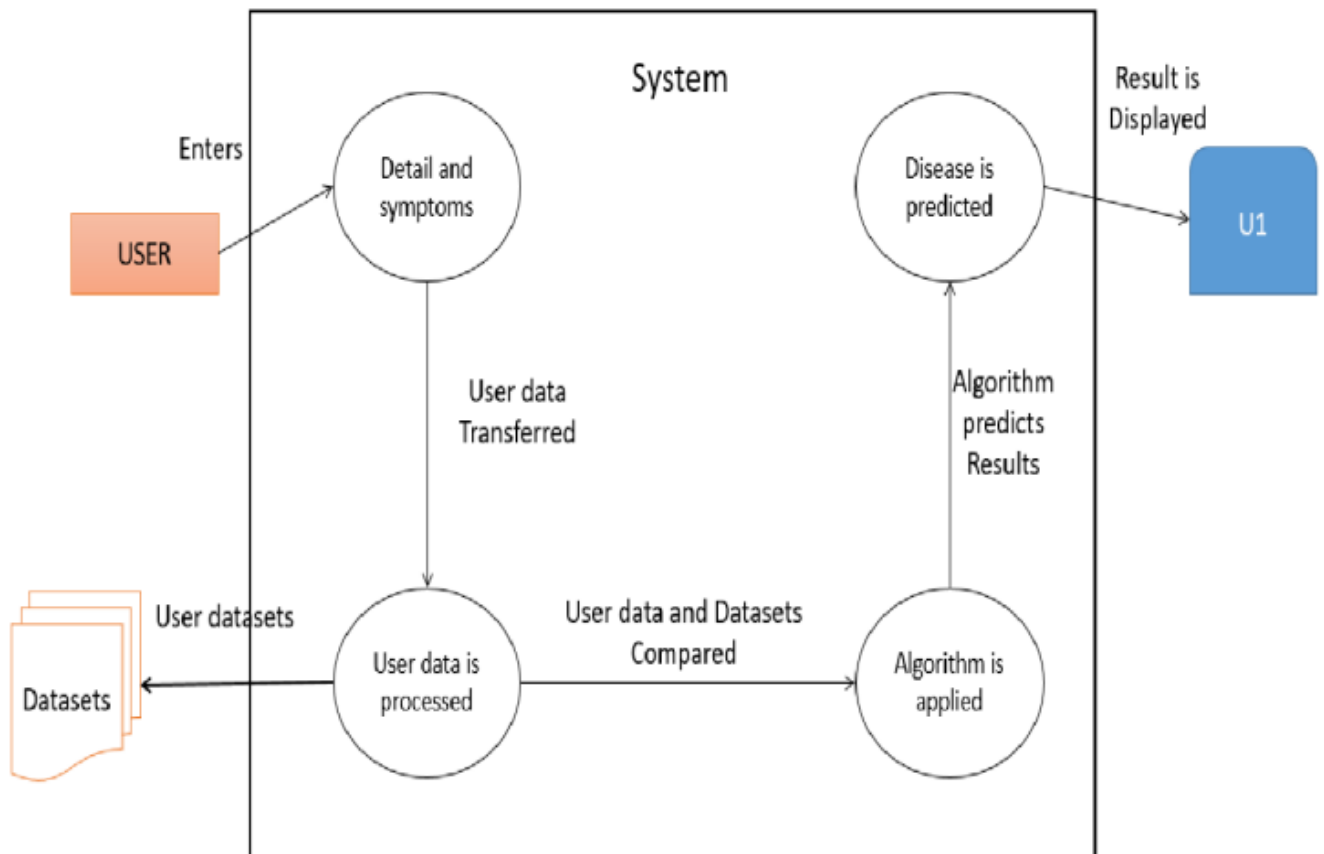


COLLABORATIVE DIAGRAM

### 5.1.7 DATA FLOW DIAGRAM:

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.



**DATA FLOW DIAGRAM**

## **CHAPTER: 6 MODULES**

## **6.1 MODULES NAME:**

- Data Preprocessing Module
- Feature Selection Module
- Module Training Module
- Model Evaluation Module
- Disease Prediction Module
- Data collection Module
- User Interface Module

## **6.2 MODULE EXPLANATION**

### **1. Data Preprocessing Module:**

- Cleans and preprocesses the collected data to make it suitable for machine learning algorithms.
- Handles missing values by imputation or removal.
- Identifies and handles outliers.
- Performs data normalization or standardization to scale the data

### **2. Feature Selection Module:**

- Identifies and extracts relevant features from the preprocessed data.
- Creates new features based on existing ones.
- Performs feature transformation such as encoding categorical variables, scaling numerical features, and handling date-time data.

### **3. Model Training Module:**

- Trains machine learning models using the preprocessed and selected features.
- Experiments with various algorithms such as decision trees, random forests, support vector machines, and artificial neural networks.

- Uses techniques such as grid search and cross-validation to optimize hyperparameters.

#### **4. Model Evaluation Module:**

- Evaluating the performance of trained models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
- Performing cross-validation to ensure the robustness of the models.

#### **5. Disease Prediction Module:**

- Uses the trained machine learning models to predict the likelihood of a patient developing a particular disease.
- Provides prediction results to the user based on input data.
- May include post-processing steps such as probability calibration.

#### **6. Data Collection module:**

- This module is responsible for collecting patient data from various sources such as hospitals, clinics, or health records.
- It may involve retrieving data from databases, files, or APIs.
- Data collected may include patient demographics, medical history, laboratory test results, and other relevant information

#### **7. User Interface Module:**

- Provides an interface for users to interact with the system.
- Collects input data from users through forms or file uploads.
- Displays prediction results to users in an understandable format such as tables, charts, or visualizations.

#### **8. Reporting Module:**

- Generates reports and visualizations to present the prediction results to stakeholders.
- Creates interactive dashboards for data analysis using tools such as Jupyter Notebooks, Microsoft Power BI, or Tableau.

## **6.3 ALGORITHM:**

### **1. DECISION TREES:**

Decision trees are a simple and widely used classification algorithm. They partition the feature space into smaller regions based on feature values. At each node of the tree, the algorithm selects the best split based on a criterion such as Gini impurity or information gain. This process continues recursively until a stopping criterion is met, such as reaching a maximum depth or when further splitting does not improve the model's performance.

#### **Advantages:**

- Easy to interpret and visualize.
- Can handle both numerical and categorical data.
- Requires little data preprocessing.

### **2. RANDOM FORESTS:**

Random forests are an ensemble learning method based on decision trees. They construct multiple decision trees during training and output the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Random forests reduce overfitting by averaging multiple decision trees trained on different subsets of the data.

#### **Advantages:**

- Reduced overfitting compared to individual decision trees.
- Robust to noise and outliers.
- Can handle large datasets with high dimensionality.

### **3. SUPPORT VECTOR MACHINES (SVM):**

SVM is a powerful supervised learning algorithm used for classification and regression tasks. SVM finds the optimal hyperplane that best separates the classes in the feature space. SVM can handle both linearly separable and non-linearly separable data using the kernel trick.

**Advantages:**

- Effective in high-dimensional spaces.
- Robust to overfitting, especially in high-dimensional space.
- Versatile: Various kernel functions can be used for different types of data.

**4. ARTIFICIAL NEURAL NETWORKS (ANN):**

ANN is a computational model inspired by the structure and functioning of the human brain. ANNs consist of interconnected nodes organized in layers, and each layer performs specific operations on the input data. ANNs are capable of learning complex patterns and relationships in data and are widely used for classification and regression tasks.

**Advantages:**

- Ability to learn complex patterns and relationships in data.
- Non-linear decision boundaries can be learned.
- Robust to noisy data.

## **CHAPTER: 7 SOFTWARE SPECIFICATION**



## 7.1 SOFTWARE SPECIFICATION:

The disease prediction system is developed using Python programming language (version 3.6 or later) and requires the following software environment for development, deployment, and execution:

## 7.2 OPERATING SYSTEM:

The system is compatible with Windows 7 or later, macOS 10.12 (Sierra) or later, and Linux (Ubuntu 16.04 or later) operating systems, providing flexibility in deployment across different platforms.

- Windows 7 or later
- macOS 10.12 (Sierra) or later
- Linux (Ubuntu 16.04 or later)

## 7.3 PROGRAMMING LANGUAGES:

- Python 3.6 or later

## 7.4 INTEGRATED DEVELOPMENT ENVIRONMENT (IDE):

- **Jupyter Notebook:** Jupyter Notebook provides an interactive computing environment for developing and documenting code. It allows the creation and sharing of documents containing live code, equations, visualizations, and narrative text.
- **PyCharm:** PyCharm is a powerful IDE for Python development, offering smart code assistance, code navigation, and integrated tools for efficient coding, testing, and debugging.
- **Spyder:** Spyder is an open-source IDE designed for scientific computing and data analysis. It provides features such as an interactive console, variable explorer, and integrated help system.
- **Visual Studio Code:** Visual Studio Code is a lightweight yet powerful code editor

that supports various programming languages, including Python. It offers features such as debugging, syntax highlighting, and version control integration.

## 7.5 MACHINE LEARNING LIBRARIES:

- **NumPy:** NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- **pandas:** pandas is a powerful data analysis and manipulation library for Python. It offers data structures such as DataFrame and Series, along with tools for reading and writing data from various file formats.
- **scikit-learn:** scikit-learn is a machine learning library for Python that provides simple and efficient tools for data mining and data analysis. It includes various algorithms for classification, regression, clustering, and dimensionality reduction.
- **TensorFlow or PyTorch:** TensorFlow and PyTorch are deep learning frameworks that provide a flexible and efficient platform for building and training deep learning models. They offer high-level APIs for building neural networks and low-level APIs for customization and experimentation.
- **Keras:** Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, Theano, or CNTK. It allows for easy and fast prototyping of deep learning models.

## 7.6 DATABASE MANAGEMENT SYSTEM (OPTIONAL):

**PostgreSQL:** PostgreSQL is a powerful, open-source object-relational database system known for its reliability, robustness, and support for advanced features such as transactions, foreign keys, and views.

**MySQL:** MySQL is an open-source relational database management system that is widely used for web-based applications. It is known for its speed, reliability, and ease of use.

**SQLite:** SQLite is a lightweight, serverless, self-contained SQL database engine that is

embedded into the application. It is often used for development and testing purposes due to its simplicity and ease of use.

## **7.7 VERSION CONTROL SYSTEM:**

**Git:** Git is a distributed version control system that allows multiple developers to collaborate on a project efficiently. It provides features such as branching, merging, and version history tracking, enabling developers to work on different features simultaneously and merge their changes seamlessly.

## **CHAPTER: 8 IMPLEMENTATION**

## SOURCE CODE:

```
from tkinter import *

import numpy as np

import pandas as pd

#symptoms

L1=['back_pain','constipation','abdominal_pain','diarrhoea','mild_fever','yellow_urine','yellowing_
of_eyes','acute_liver_failure','fluid_overload','swelling_of_stomach','swelled_lymph_nodes','malai
se','blurred_and_distorted_vision','phlegm','throat_irritation','redness_of_eyes','sinus_pressure','run
ny_nose','congestion','chest_pain','weakness_in_limbs','fast_heart_rate','pain_during_bowel_move
ments','pain_in_anal_region','bloody_stool','irritation_in_anus','neck_pain','dizziness','cramps','bru
ising','obesity','swollen_legs','swollen_blood_vessels','puffy_face_and_eyes','enlarged_thyroid','bri
ttle_nails','swollen_extremeties','excessive_hunger','extra_marital_contacts','drying_and_tingling_
lips','slurred_speech','knee_pain','hip_joint_pain','muscle_weakness','stiff_neck','swelling_joints','
movement_stiffness','spinning_movements','loss_of_balance','unsteadiness','weakness_of_one_bo
dy_side','loss_of_smell','bladder_discomfort','foul_smell_ofurine','continuous_feel_of_urine','pass
age_of_gases','internal_itching','toxic_look_(typhos)','depression','irritability','muscle_pain','altere
d_sensorium','red_spots_over_body','belly_pain','abnormal_menstruation','dischromic_patches','w
atering_from_eyes','increased_appetite','polyuria','family_history','mucoid_sputum','rusty_sputum'
,'lack_of_concentration','visual_disturbances','receiving_blood_transfusion','receiving_unsterile_i
njections','coma','stomach_bleeding','distention_of_abdomen','history_of_alcohol_consumption','fl
uid_overload','blood_in_sputum','prominent_veins_on_calf','palpitations','painful_walking','pus_fi
lled_pimples','blackheads','scurring','skin_peeling','silver_like_dusting','small_dents_in_nails','infl
```

```
ammatory_nails','blister','red_sore_around_nose','yellow_crust_ooze']
```

```
#Disease Names
```

```
disease=['Fungal infection','Allergy','GERD','Chronic cholestasis','Drug Reaction','Peptic ulcer disease','AIDS','Diabetes','Gastroenteritis','Bronchial Asthma','Hypertension','Migraine','Cervical spondylosis','Paralysis(brainhemorrhage)','Jaundice','Malaria','Chickenpox','Dengue','Typhoid','hepatitisA','HepatitisB','HepatitisC','HepatitisD','HepatitisE','Alcoholichepatitis','Tuberculosis','CommonCold','Pneumonia','Dimorphichemorrhoids(piles)','Heartattack','Varicoseveins','Hypothyroidism','Hyperthyroidism','Hypoglycemia','Osteoarthritis','Arthritis','(vertigo)Paroysmal Positional Vertigo','Acne','Urinary tract infection','Psoriasis','Impetigo']
```

```
l2=[]
```

```
for x in range(0,len(l1)):
```

```
    l2.append(0)
```

```
# TESTING DATA df -----
```

```
df=pd.read_csv("Training.csv")
```

```
df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug Reaction':4,
```

```
'Peptic ulcer disease':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asthma':9,'Hypertension ':10,
```

```
'Migraine':11,'Cervical spondylosis':12,'Paralysis (brain
```

```
hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Dengue':17,'Typhoid':18,'hepatitis
```

```
A':19,'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis
```

```
E':23,'Alcoholichepatitis':24,'Tuberculosis':25,'CommonCold':26,'Pneumonia':27,'Dimorphichem
```

morhoids

(piles)':28,'Heart

attack':29,'Varicoseveins':30,'Hypothyroidism':31,'Hyperthyroidism':32,'Hypoglycemia':

33,'Osteoarthritis':34,'Arthritis':35,'(vertigo) Paroysmal Positional Vertigo':36,'Acne':37,'Urinary

tract infection':38,'Psoriasis':39,'Impetigo':40} },inplace=True)

# print(df.head())

X= df[11]

y = df[["prognosis"]]

np.ravel(y)

# print(y)

# TRAINING DATA tr -----

tr=pd.read\_csv("Testing.csv")

tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug

Reaction':4,'Pepticulcer disease':5,'AIDS':6,'Diabetes':7,'Gastroenteritis':8,'Bronchial

Asthma':9,'Hypertension':10,'Migraine':11,'Cervical spondylosis':12,'Paralysis (brain

haemorrhage)':13,'Jaundice':14,'Malaria':15,'Chickenpox':16,'Dengue':17,'Typhoid':

18,'hepatitis A':19,'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic

hepatitis':24,'Tuber

culosis':25,'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart

attack':29,'Varicose veins':

30,'Hypothyroidism':31,'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthritis':34,'Arthritis':35,'

```
(vertigo)          Paroysmal          Positional          Vertigo':36,'Acne':37,'Urinary          tract
infection':38,'Psoriasis':39,'Impetigo':40} },inplace=True)
```

```
X_test= tr[11]
```

```
y_test = tr[["prognosis"]]
```

```
np.ravel(y_test)
```

```
# -----
```

```
def DecisionTree():
```

```
    from sklearn import tree
```

```
    clf3 = tree.DecisionTreeClassifier() # empty model of the decision tree
```

```
    clf3 = clf3.fit(X,y)
```

```
    # calculating accuracy-----
```

```
    from sklearn.metrics import accuracy_score
```

```
    y_pred=clf3.predict(X_test)
```

```
    print(accuracy_score(y_test, y_pred))
```

```
    print(accuracy_score(y_test, y_pred,normalize=False))
```

```
    # -----
```

```
psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
```



```

for k in range(0,len(l1)):

    # print (k,)

    for z in psymptoms:

        if(z==l1[k]):

            l2[k]=1

    inputtest = [l2]

predict = clf3.predict(inputtest)

predicted=predict[0]


h='no'

for a in range(0,len(disease)):

    if(predicted == a):

        h='yes'

        break


if (h=='yes'):

    t1.delete("1.0", END)

    t1.insert(END, disease[a])

else:

    t1.delete("1.0", END)

    t1.insert(END, "Not Found")

def randomforest():

```

```

from sklearn.ensemble import RandomForestClassifier

clf4 = RandomForestClassifier()

clf4 = clf4.fit(X,np.ravel(y))


# calculating accuracy-----

from sklearn.metrics import accuracy_score

y_pred=clf4.predict(X_test)

print(accuracy_score(y_test, y_pred))

print(accuracy_score(y_test, y_pred,normalize=False))

# -----

psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]


for k in range(0,len(l1)):

    for z in psymptoms:

        if(z==l1[k]):

            l2[k]=1

inputtest = [l2]

predict = clf4.predict(inputtest)

predicted=predict[0]


h='no'

for a in range(0,len(disease)):

    if(predicted == a):

```

```
h='yes'
```

```
break
```

```
if (h=='yes'):
```

```
    t2.delete("1.0", END)
```

```
    t2.insert(END, disease[a])
```

```
else:
```

```
    t2.delete("1.0", END)
```

```
    t2.insert(END, "Not Found")
```

```
def NaiveBayes():
```

```
from sklearn.naive_bayes import GaussianNB
```

```
gnb = GaussianNB()
```

```
gnb=gnb.fit(X,np.ravel(y))
```

```
# calculating accuracy-----
```

```
from sklearn.metrics import accuracy_score
```

```
y_pred=gnb.predict(X_test)
```

```
print(accuracy_score(y_test, y_pred))
```

```
print(accuracy_score(y_test, y_pred,normalize=False))
```

```
# -----
```

```
psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
```

```
for k in range(0,len(l1)):
```

```

for z in psymptoms:

    if(z==l1[k]):

        l2[k]=1

inputtest = [l2]

predict = gnb.predict(inputtest)

predicted=predict[0]


h='no'

for a in range(0,len(disease)):

    if(predicted == a):

        h='yes'

        break


if (h=='yes'):

    t3.delete("1.0", END)

    t3.insert(END, disease[a])

else:

    t3.delete("1.0", END)

    t3.insert(END, "Not Found")


root = Tk()

root.configure(background='blue')

# entry variables

```

```
Symptom1 = StringVar()
```

```
Symptom1.set(None)
```

```
Symptom2 = StringVar()
```

```
Symptom2.set(None)
```

```
Symptom3 = StringVar()
```

```
Symptom3.set(None)
```

```
Symptom4 = StringVar()
```

```
Symptom4.set(None)
```

```
Symptom5 = StringVar()
```

```
Symptom5.set(None)
```

```
Name = StringVar()
```

```
# Heading
```

```
w2 = Label(root, justify=LEFT, text="DISEASE PREDICTION USING MACHINE  
LEARNING", fg="white", bg="blue")
```

```
w2.config(font=("Elephant", 30))
```

```
w2.grid(row=1, column=0, columnspan=2, padx=100)
```

```
w2.config(font=("Aharoni", 30))
```

```
w2.grid(row=2, column=0, columnspan=2, padx=100)
```

```
# labels
```

```
NameLb = Label(root, text="Name of the Patient", fg="yellow", bg="black")
```

```
NameLb.grid(row=6, column=0, pady=15, sticky=W)
```

```
S1Lb = Label(root, text="Symptom 1", fg="yellow", bg="black")
```

```
S1Lb.grid(row=7, column=0, pady=10, sticky=W)
```

```
S2Lb = Label(root, text="Symptom 2", fg="yellow", bg="black")
```

```
S2Lb.grid(row=8, column=0, pady=10, sticky=W)
```

```
S3Lb = Label(root, text="Symptom 3", fg="yellow", bg="black")
```

```
S3Lb.grid(row=9, column=0, pady=10, sticky=W)
```

```
S4Lb = Label(root, text="Symptom 4", fg="yellow", bg="black")
```

```
S4Lb.grid(row=10, column=0, pady=10, sticky=W)
```

```
S5Lb = Label(root, text="Symptom 5", fg="yellow", bg="black")
```

```
S5Lb.grid(row=11, column=0, pady=10, sticky=W)
```

```
lrLb = Label(root, text="DecisionTree", fg="white", bg="red")
```

```
lrLb.grid(row=15, column=0, pady=10, sticky=W)
```

```
destreeLb = Label(root, text="RandomForest", fg="white", bg="red")
```

```
destreeLb.grid(row=17, column=0, pady=10, sticky=W)
```

```
ranfLb = Label(root, text="NaiveBayes", fg="white", bg="red")
```

```
ranfLb.grid(row=19, column=0, pady=10, sticky=W)
```

```
# entries
```

```
OPTIONS = sorted(l1)
```

```
NameEn = Entry(root, textvariable=Name)
```

```
NameEn.grid(row=6, column=1)
```

```
S1En = OptionMenu(root, Symptom1,*OPTIONS)
```

```
S1En.grid(row=7, column=1)
```

```
S2En = OptionMenu(root, Symptom2,*OPTIONS)
```

```
S2En.grid(row=8, column=1)
```

```
S3En = OptionMenu(root, Symptom3,*OPTIONS)
```

```
S3En.grid(row=9, column=1)
```

```
S4En = OptionMenu(root, Symptom4,*OPTIONS)
```

```
S4En.grid(row=10, column=1)
```

```
S5En = OptionMenu(root, Symptom5,*OPTIONS)
```

```
S5En.grid(row=11, column=1)
```

```
dst = Button(root, text="DecisionTree", command=DecisionTree,bg="green",fg="yellow")
```

```
dst.grid(row=8, column=3,padx=10)
```

```
rnf = Button(root, text="Randomforest", command=randomforest,bg="green",fg="yellow")
```

```
rnf.grid(row=9, column=3,padx=10)
```

```
lr = Button(root, text="NaiveBayes", command=NaiveBayes,bg="green",fg="yellow")
```

```
lr.grid(row=10, column=3,padx=10)
```

```
#textfileds
```

```
t1 = Text(root, height=1, width=40,bg="orange",fg="black")
```

```
t1.grid(row=15, column=1, padx=10)
```

```
t2 = Text(root, height=1, width=40,bg="orange",fg="black")
```

```
t2.grid(row=17, column=1 , padx=10)
```

```
t3 = Text(root, height=1, width=40,bg="orange",fg="black")
```

```
t3.grid(row=19, column=1 , padx=10)
```

```
root.mainloop()
```



## **CHAPTER : 9 SOFTWARE TESTING**

## **9.1. FEASIBILITY STUDY:**

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.

In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

They are 3 types of Feasibility

- Economical feasibility
- Technical feasibility
- Operational feasibility
- Social feasibility

### **9.1.1. ECONOMICAL FEASIBILITY :**

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not.

### **9.1.2. TECHNICAL FEASIBILITY:**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

### **9.1.3 OPERATIONAL FEASIBILITY:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

### **9.1.4 SOCIAL FEASIBILITY:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

## **9.2. SYSTEM TESTING:**

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases makes it shine.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **9.2.1. VARIOUS LEVELS OF TESTING:**

- 1. Unit Testing**
- 2. Integration Testing**
- 3. System Testing**
- 4. User Acceptance Testing**
- 5. Regression Testing**
- 6. Performance Testing**

#### **9.2.1.1. UNIT TESTING:**

This type of testing is focused on testing individual units or components of the software, such as functions or methods. It can help to ensure that each unit of the system is working as intended. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

## **2. INTEGRATION TESTING:**

This type of testing is focused on testing how different components of the system work together. It can help to identify any issues or bugs that may arise when different parts of the system interact with each other. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## **3. SYSTEM TESTING:**

This type of testing is focused on testing the entire system as a whole, including all components and interactions between them. It can help to ensure that the system meets all requirements and specifications, and is functioning as intended. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## **4. FUNCTIONAL TESTING:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes,

and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## **5. USER ACCEPTANCE TESTING:**

This type of testing involves getting feedback from actual users of the system, to ensure that it meets their needs and is easy to use. It can help to identify any usability issues or areas where the system may need improvement.

## **6. REGRESSION TESTING:**

This type of testing involves retesting previously tested functionality after changes have been made to the system, to ensure that the changes did not introduce any new issues or bugs.

## **7. PERFORMANCE TESTING:**

This type of testing is focused on testing the system's performance under various conditions, such as high traffic or heavy loads. It can help to identify any areas where the system may need optimization or improvement.

## **8. WHITE-BOX TESTING:**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

## **9. BLACK-BOX TESTING:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be

written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box . you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

## **10. ACCEPTANCE TESTING:**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

## **CHAPTER: 10 OUTPUT SCREENSHOTS**



## Output:

**Disease Predictor using Machine Learning**

Name of the Patient

Symptom 1

Symptom 2

Symptom 3

Symptom 4

Symptom 5

DecisionTree

Randomforest

NaiveBayes

DecisionTree

RandomForest

NaiveBayes

# Disease Predictor using Machine Learning

Name of the Patient

Vamsi

### Symptom 1

abdominal\_pain ==

## Symptom 2

back\_pain —

## DecisionTree

### Symptom 3

chest\_pain

## Randomforest

### Symptom 4

dischromic\_patches —

NaiveBayes

### Symptom 5

irritability —

## DecisionTree

GERD

## RandomForest

Fungal infection

## NaiveBayes

GERD

## **CHAPTER: 11 CONCLUSION & FUTURE ENHANCEMENT**

## 11.1 CONCLUSION:

The disease prediction system developed using machine learning approaches provides a comprehensive solution for predicting the likelihood of patients developing specific diseases. By leveraging Python programming language and a variety of machine learning libraries, including scikit-learn, TensorFlow, and Keras, the system can preprocess data, train machine learning models, and perform disease predictions with high accuracy.

The software environment for development, deployment, and execution is versatile, supporting multiple operating systems and providing a range of integrated development environments (IDEs) such as Jupyter Notebook, PyCharm, Spyder, and Visual Studio Code. This flexibility allows developers to choose the tools that best suit their preferences and workflow.

Additionally, the system incorporates data visualization libraries such as Matplotlib and Seaborn for creating informative and visually appealing plots and charts. Optional integration with web development frameworks like Flask or Django and database management systems such as PostgreSQL, MySQL, or SQLite provides scalability and data management capabilities for larger-scale deployments.

Version control using Git ensures collaboration and code management, while documenting and reporting tools such as Jupyter Notebook, Microsoft Word, or LaTeX facilitate the creation of professional-looking reports and documents. The software environment, including multiple operating system compatibility, versatile IDEs, and robust version control, ensures efficient development, deployment, and maintenance of the system.

In conclusion, the disease prediction system, with its robust software environment and machine learning capabilities, offers an effective solution for healthcare professionals and researchers to predict and prevent the onset of diseases, ultimately improving patient outcomes and quality of care.

## **11.2 FUTURE ENHANCEMENTS:**

While the proposed system is a significant step forward in attendance management, there are several areas for future enhancements and improvements:

### **1. Integration of Additional Data Sources:**

- Incorporate additional data sources such as genetic information, environmental factors, and lifestyle data to improve prediction accuracy and robustness.

### **2. Advanced Feature Engineering Techniques:**

- Explore advanced feature engineering techniques such as feature synthesis, interaction features, and domain-specific feature extraction to capture complex relationships in the data.

### **3. Ensemble Learning Methods:**

- Implement ensemble learning methods such as stacking, blending, and boosting to further improve prediction performance by combining the strengths of multiple machine learning models.

### **4. Deep Learning Architectures:**

- Experiment with more complex deep learning architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer models, for better feature representation and prediction.

### **5. Real-time Monitoring and Alerts:**

- Develop real-time monitoring capabilities to continuously update prediction models and provide timely alerts to healthcare providers and patients about potential disease risks.

### **6. Interpretability and Explainability:**

- Enhance model interpretability and explainability using techniques such as SHAP (SHapley Additive exPlanations) values, LIME (Local Interpretable Model-agnostic Explanations), and integrated gradients.

### **7. Deployment as a Web Application:**

- Deploy the system as a web application with an intuitive user interface, allowing healthcare professionals and patients to access predictions, visualize results, and track health status conveniently.

### **8. Integration with Electronic Health Records (EHRs):**

- Integrate the system with electronic health record (EHR) systems to streamline data access, ensure data privacy and security, and facilitate seamless information exchange between healthcare providers.

#### **9. Continuous Model Improvement:**

- Implement automated model retraining and updating pipelines to incorporate new data, adapt to changing patient populations, and continuously improve prediction accuracy over time.

#### **10. Collaboration with Research Institutions:**

- Collaborate with research institutions and healthcare organizations to validate the system's performance on large-scale, diverse datasets and ensure its effectiveness across different demographics and medical conditions.

### **11.3 APPLICATION:**

#### **1. Early Disease Detection:**

- The system can help in the early detection of diseases such as diabetes, heart disease, cancer, and neurological disorders by analyzing patient data and identifying patterns indicative of disease onset.

#### **2. Personalized Medicine:**

- By predicting disease risks for individual patients based on their medical history, genetic information, and lifestyle factors, the system enables personalized treatment plans and preventive interventions tailored to each patient's specific needs.

#### **3. Population Health Management:**

- Healthcare providers and policymakers can use the system to analyze population health trends, identify high-risk groups, and develop targeted interventions and public health campaigns to prevent the spread of diseases and improve overall health outcomes.

## **CHAPTER: 12 REFERENCES**

## REFERENCES

- [1] C. Chauhan, et al., "Multiple Disease Prediction Using Machine Learning Algorithms," 2021.
- [2] A. Kamboj, et al., "A Machine Learning Model for Early Prediction of Multiple Diseases to Cure Lives," 2020.
- [3] S. Kolli, et al., "Symptoms Based Multiple Disease Prediction Model using Machine Learning Approach," 2021.
- [4] P. Krishnaiah, et al., "Predictive Modeling for Multiple Diseases Using Machine Learning with Feature Engineering," 2015.
- [5] H. Al-Mallah, et al., "Multiple Disease Prediction Using Hybrid Deep Learning Architecture," 2016.
- [6] Y. Gamo, et al., "Machine Learning Based Clinical Decision Support Systems for Multi-Disease Prediction: A Review," 2020.
- [7] W. Li, et al., "Towards Multi-Disease Prediction Using Graph Neural Networks," 2020.
- [8] R. Ribeiro, et al., "A Survey on Explainable AI Techniques for Diagnosis and Prognosis in Healthcare," 2020.
- [9] E. Char, et al., "Ethical Considerations in AI-Driven Healthcare," 2020. M. Wild, et al., "Streamlit for Machine Learning: Creating Interactive Web Apps in Python," 2022.
- [10] Sayantan Saha, Argha Roy Chowdhuri et al, "Web Based Disease Detection System",IJERT, ISSN:22780181, Vol.2 Issue 4, April-2013
- [11] Shadab Adam et al "Prediction system for Heart Disease using Naïve Bayes", International



Journal of advanced Computer and Mathematical Sciences, ISSN 2230- 9624, Vol 3,Issue 3,2012,pp 290- 294[Accepted- 12/06/2012].

[12]Min Chen, Yixue Hao et.al “Disease Prediction by Machine Learning over big data from Healthcare Communities”, IEEE[Access 2017]

[13]Mr Chintan Shah,Dr. Anjali Jivani, “Comparison Of Data Mining Classification Algorithms for Breast Cancer Prediction”, IEEE-31661

[14]Palli Suryachandra, Prof.Venkata Subba Reddy,“Comparison of Machine Learning algorithms For Breast Cancer”, IEEE.

[15]Andrew Alikberov, Stephan Broadly et.al “The Learning Machine”, Accessed on: March 26,2020. [Online]. Available: <https://www.thelearningmachine.ai>



# VELAMMAL ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## NCAIF'24



INSTITUTION'S  
INNOVATION  
COUNCIL  
(Ministry of HRD Initiative)



### CERTIFICATE OF PARTICIPATION



This is to certify that Dr./Mr./Ms. Usha. G. B

of T.J.S Engineering College has presented a

paper titled Speech Emotion Recognition

\_\_\_\_\_ in the

**9th National Conference on Artificial Intelligence Frontiers: Shaping the Future of Technology** held on 12th March, 2024.

**HOD-CSE**

Dr. B. MURUGESHWARI



ICTACADEMY®

**Principal**

Dr. S. SATISH KUMAR



# VELAMMAL ENGINEERING COLLEGE

AN AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## NCAIF'24



INSTITUTION'S  
INNOVATION  
COUNCIL  
(Ministry of HRD Initiative)



### CERTIFICATE OF PARTICIPATION



This is to certify that ~~Dr./Mr./Ms.~~ Manugunta Devi prasanthi  
of T.J.S Engineering College has presented a  
paper titled Speech Emotion Recognition

\_\_\_\_\_ in the  
**9th National Conference on Artificial Intelligence Frontiers: Shaping  
the Future of Technology** held on 12th March, 2024.

**HOD-CSE**

Dr. B. MURUGESHWARI



ICTACADEMY®

**Principal**

Dr. S. SATISH KUMAR





**VELAMMAL ENGINEERING COLLEGE**

AN AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**NCAIF'24**



**CERTIFICATE OF PARTICIPATION**



This is to certify that Dr./Mr./Ms. Vunnam Santhosh  
of T.J.S Engineering College has presented a  
paper titled Speech Emotion Recognition  
\_\_\_\_\_ in the  
**9th National Conference on Artificial Intelligence Frontiers: Shaping  
the Future of Technology** held on 12th March, 2024.

**HOD-CSE**

*Dr. B. MURUGESHWARI*



**ICTACADEMY®**

**Principal**

*Dr. S. SATISH KUMAR*