VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi ,Karnataka,INDIA

A Project Report
On

Expert System for Differential Diagnosis using Deep Learning

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted By

PRAVEEN V	**1GA16CS101**
YASHAS C R	**1GA16CS182**
THEJASVEE M	**1GA16CS195**
SNEHA SURENDRA	**1GA16CS198**

Under the Guidance of

Mrs. JYOTHI R
Assistant Professor

Department of Computer Science and Engineering
GLOBAL ACADEMY OF TECHNOLOGY
Accredited by NBA(2019-2022)
Rajarajeshwari nagar, Bengaluru - 560 098

2019 – 2020

GLOBAL ACADEMY OF TECHNOLOGY

Accredited by NBA(2019-2022)

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Project Entitled **"*Expert* System *for differential diagnosis using deep learning*"** carried out by **PRAVEEN V**, bearing **USN 1GA16CS101, YASHAS C R**, bearing **USN 1GA16CS182, THEJASVEE M**, bearing **USN 1GA16CS195, SNEHA SURENDRA**, bearing **USN 1GA16CS198,** bonafide student of Global Academy of Technology, is in partial fulfillment for the award of the **BACHELOR OF ENGINEERING** in Computer Science and Engineering from **Visvesvaraya Technological University, Belagavi** during the year 2019-2020. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report submitted in the department library. The Project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

Jyothi R Assistant Professor Dept. of CSE GAT, Bengaluru.	Dr. N Guruprasad Professor Dept. of CSE GAT, Bengaluru.

Name of the Examiners	Signature with date
1. ________________	________________
2. ________________	________________

GLOBAL ACADEMY OF TECHNOLOGY

Accredited by NBA(2019-2022)

Rajarajeshwari nagar, Bengaluru – 560 098

DECLARATION

We, **PRAVEEN V**, bearing bearing **USN 1GA16CS101, YASHAS C R**, bearing **USN 1GA16CS182, THEJASVEE M**, bearing **USN 1GA16CS195, SNEHA SURENDRA**, bearing **USN 1GA16CS198,** students of Seventh Semester B.E, Department of Computer Science and Engineering, Global Academy of Technology, Rajarajeshwari nagar Bengaluru, declare that the Project Work entitled "**Expert System for differential diagnosis using deep learning"** has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in **Bachelor of Engineering** in **Computer Science and Engineering** from **Visvesvaraya Technological University, Belagavi** during the academic year **2019-2020**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree.

PRAVEEN V	1GA16CS101
YASHAS C R	1GA16CS182
THEJASVEE M	1GA16CS195
SNEHA SURENDRA	1GA16CS198

Place: Bengaluru

Date: 18/08/20

ABSTRACT

Differential Diagnosis is a process of differentiating two or more condition which shares similar signs or symptoms. This is based on a knowledge of the pathophysiology of the presenting signs and symptoms, and the natural history of various diseases and their causative agents. This process becomes cumbersome when the symptoms and patients history is complex. This challenges faced in the differential diagnosis are diagnosing the disease of the patients golden time and also keeping in account the vast knowledge of the symptoms.

We aim to tackle the poor rate of diagnosing, finding the accurate disease and commencing the treatment. Due to the lack of knowledge and loss of information there's been a negative diagnosis. We focus on utilizing the golden hours for diagnosing rather on treatment.

Considering the above challenges in differential diagnosis we try to overcome them using machine learning. We create a knowledge base required for diagnosing the diseases and collect the patients' data such as symptoms, history, records and map it to the most possible diseases. Thus helps in diagnosing faster and narrows down the focus area.

The main methodology implemented in this project after collection and labelling of data is data pre-processing, attributes selection and finally classification of this data by KNN, ANN Back Propagation Algorithm in Machine Learning. The MLP, which is a deep learning method, uses back propagation for training the network which helps in characterizing several layers of input nodes connected as a directed graph between the input and output layers.

As an end result of the training module, we display the disease with its probability with a brief description of the disease along with the possible symptoms which may further occur. With the result of highest accuracy that is obtained in the shortest implementation time, we aspire to do a small contribution to the medical field.

ACKNOWLEDGEMENT

The satisfaction and the euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. The constant guidance of these persons and encouragement provide, crowned our efforts with success and glory. Although it is not possible to thank all the members who helped for the completion of the project individually, we take this opportunity to express our gratitude to one and all.

We are grateful to management and our institute **GLOBAL ACADEMY OF TECHNOLOGY** with its very ideals and inspiration for having provided us with the facilities, which made this, project a success.

We express our sincere gratitude to **Dr. N. Rana Pratap Reddy,** Principal, Global Academy of Technology for the support and encouragement.

We wish to place on record, our grateful thanks to **Dr. N Guruprasad**, Professor, Department of CSE, Global Academy of Technology, for the constant encouragement provided to us.

We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance and valid suggestion given to us by our guide **Mrs. Jyothi R, Assistant Professor**, Department of CSE, Global Academy of Technology.

We are thankful to all the staff members of the department for providing relevant information and helped in different capacities in carrying out this project.

Last, but not least, we owe our debts to our parents, friends and also those who directly or indirectly have helped us to make the project work a success.

PRAVEEN V	**1GA16CS101**
YASHAS C R	**1GA16CS182**
THEJASVEE M	**1GA16CS195**
SNEHA SURENDRA	**1GA16CS198**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AI	Artificial Intelligence
ML	Machine Learning
MLP	Multi-Layer Perceptron
KNN	K- Nearest Neighbor
ANN	Artificial Neural Network
SRS	Software Requirement Specification
DFD	Data Flow Diagram

Chapter 1

INTRODUCTION

Machine learning is a branch of artificial intelligence that aims at solving real life engineering problems. It provides the opportunity to learn without being explicitly programmed and it is based on the concept of learning from data. It is so much ubiquitously used dozen a times a day that we may not even know it. The advantage of machine learning (ML) methods is that it uses mathematical models, heuristic learning, knowledge acquisitions and decision trees for decision making. Thus, it provides controllability, observability and stability. It updates easily by adding a new patient‘s record.

The amount of Medical data recorded in hospitals and its significance as an ever-growing source of information has been long known and proven. Though the importance of the information hidden in these records has never been doubted, this data has mostly been used only for clinical purposes. Only recently has this been properly mined for valuable information to be used for research and to develop systems that assist the medical fraternity. Mostly, the systems that make use of this information are domain specific systems that predict diseases restricted to their area of specialization (like heart, brain etc.). But these systems are limited and are not applicable to the whole medical dataset. Our system uses this vast storage of information so that diagnosis based on this historical data can be made. This system aids medical diagnosis in the whole dataset by computing the probability of occurrence of a particular ailment from the medical data. The system mines the data using a unique algorithm which increases accuracy of such diagnosis by combining Neural Networks and Differential Diagnosis all integrated into one single approach. The strengths of kNN, Feed Forward algorithm, SOM, back propagation and P2P Grid Architecture are used to make the system unique and effectively enhanced symptoms.

1.1 Definitions

Machine Learning- It is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

Differential Diagnosis: In medicine, a differential diagnosis is the distinguishing of a particular disease or condition from others that present similar clinical features. Differential diagnostic procedures are used by physicians to diagnose the specific disease in a patient, or, at least, to eliminate any imminently life-threatening conditions. Often, each individual option of a possible disease is called a differential diagnosis (e.g. acute bronchitis could be a differential diagnosis in the evaluation of a cough, even if the final diagnosis is common cold).

Neural Networks- Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

Multilayer perceptron (MLP) is a feed-forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses back-propagation for training the network. MLP is a deep learning method.

1.2 Project Outline

Chapter 1: Introduction

We propose a system for diagnosing liver based on its symptoms and patients history and predict the probable diseases causing the symptoms.

Chapter 2: Survey of Literature

This system provides the interface for doctors to enter the symptoms and appropriate lab report values and diagnose the patient. Visualization of the most probable disease is shown for better understanding.

Chapter 3: System Requirement Specification

Functional and non-functional requirement that are needed in the project are listed out. Including the software and hardware requirement that includes the System, Hard disk requirement along with the operation system and languages used is listed.

Chapter 4: System Design

The overall architecture of the system, the overview of the various designs that's included in the system is shown. Data being the most integral part of project, the segregation and various level of data flow including the figures are shown. The conclusive use case diagram, activity, sequence, and class diagram is also included. The most important part of module.

Chapter 5: Implementation

Various steps of implementation and the different resources without which implementation wouldn't be possible are listed out. Some of the issues that are faced and the different algorithms used are described and its steps are shown.

Chapter 6: Testing

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Testing has been done to each module and on different criteria's ie. Unit testing, Integration testing which includes top-down and bottom-up testing, System testing.

Chapter 7: Results

Snapshots of the results of the prediction is shown. A dataset is taken and the probability of the various disease is returned. The result is shown in form of percentage of the probability.

Chapter 8: Conclusion

The conclusion of the overall report along with its future enhancement is written.

Chapter 2

REVIEW OF LITERATURE

2.1 System Study

A detailed study to determine whether, to what extent, and how automatic data-processing equipment should be used; it usually includes an analysis of the existing system and the design of the new system, including the development of system specifications which provide a basis for the selection of equipment.

2.2 Proposed Work

Our proposed system works on diagnosing diseases using the symptoms related to Liver. We present KNN, ANN Back Propagation for differential diagnosis of diseases. KNN, ANN helps in predicting the final possible diseases with recurring classification of data. The main architecture of KNN is presented, it includes the following three major phases: (1) Input of data; (2) Dataset classification; (3) Prediction of diseases.

2.2.1 Problem Statement

- Diagnosing the patient and treating for the permanent cure has been challenging forever. Patient would have undergone irreversible damage to his health condition even before identifying the actual disease he has been suffering from.
- Thus permanent cure can be achieved by a patient when he is diagnosed and treated right within the golden period of the disease.
- Developing an automated system for diagnosing the patient provided with the patient’s conditions, test results and history and listing out accurately the most appropriate diseases causing the symptoms can improve the rate of diagnosing drastically, increase the probability of permanent cure and thus saving millions of life.

2.2.2 Existing System

- Design an automated medical decision making system and help doctors in minimizing misdiagnosis.
- The database for a specific disease and related symptoms is created.
- Later proposed an algorithm to predict the Swine Flu disease on several attributes.
- They have observed the outcomes for machine learning algorithms such as Bayesian classification, Decision Tree, and compared it with the neural networks algorithms such as dynamic node creation (DNC) and feed-forward neural network construction.
- The system is more reliable and faster than the conventional systems.
- The problem statement only focuses on a single disease i.e. Swine Flu.
- Each disease will have to have its own algorithm to be precise and accurate.

2.2.3 Proposed System

We propose to design an automated diagnosing system for diagnosing various kinds of diseases from which liver can be affected. This system takes clinical data as input and gives out diagnose predicting the probable diseases causing the symptoms.

2.3 Scope of the project

Most of the time it takes a lot of time in diagnosing a disease and sometimes its even misdiagnosed. The motive of this project is to diagnose disease using symptoms with high rate of accuracy and less execution time. We hope to bridge the gap of lack of knowledge during crucial times. We’ve made use of trending technology i.e Machine Learning. We aim to collect relevant data related to our fields of study. Train this data as per our proposed algorithm and modules and predict the disease that’s causing the symptoms.

Chapter 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Functional Requirements

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the general approach taken by the organization when writing requirements. The functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on.

Functional requirements are as follows:

- Develop a robust solution which can predict the vulnerability of a disease given basic symptoms, patients history and lab reports etc.
- Train the model against enough data sets to maintain the accuracy level above 90%
- Optimize the model to rise the accuracy level even further
- Provide data visualization features for the customers to get more valuable insight of the patient health
- Expose the solution over the cloud as a service so that the solution can be re-usable by any third party applications.

3.2 Non-Functional Requirements

Nonfunctional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as capabilities of I/O devices and the data representations used in system interfaces.

The nonfunctional requirements are as follows:

- Should be easier to access it from the various browsers available.
- Response time of the applications should reflect the real time observations.
- The algorithm should never fail in any of the test cases.
- There shouldn’t be any security concerns on the merged data.
- Each user’s activity should be separated from the other user’s activities.

3.3 Hardware Requirements

- System : Intel processor with 2.2GHz and above.
- Hard Disk : 120 GB.
- Input Devices : Keyboard, Mouse
- Ram : 4 GB

3.4 Software Requirements

- Operating system : Windows 10
- Coding Language : Python
- IDE : Python IDE

Chapter 4

SYSTEM DESIGN

4.1 Design Overview

The Design Summary allows you to quickly access design overview information, reports, and messages. By default, the Design Summary appears in the Workspace when you open a project, and it displays information specific to your targeted device and software tools.

4.2 System Architecture

This figure shows the architecture of this project beginning from the user input to the output. This architecture has many stages which are as shown below. In the first stage user being a medical expert enters the appropriate data like symptoms, lab reports and patient's history. In second stage input data are then pre-processed and brought to an appropriate format. In the next stage this data is given to the trained model which will give out the set of most possible diseases causing the symptoms in the last stage on the interface.

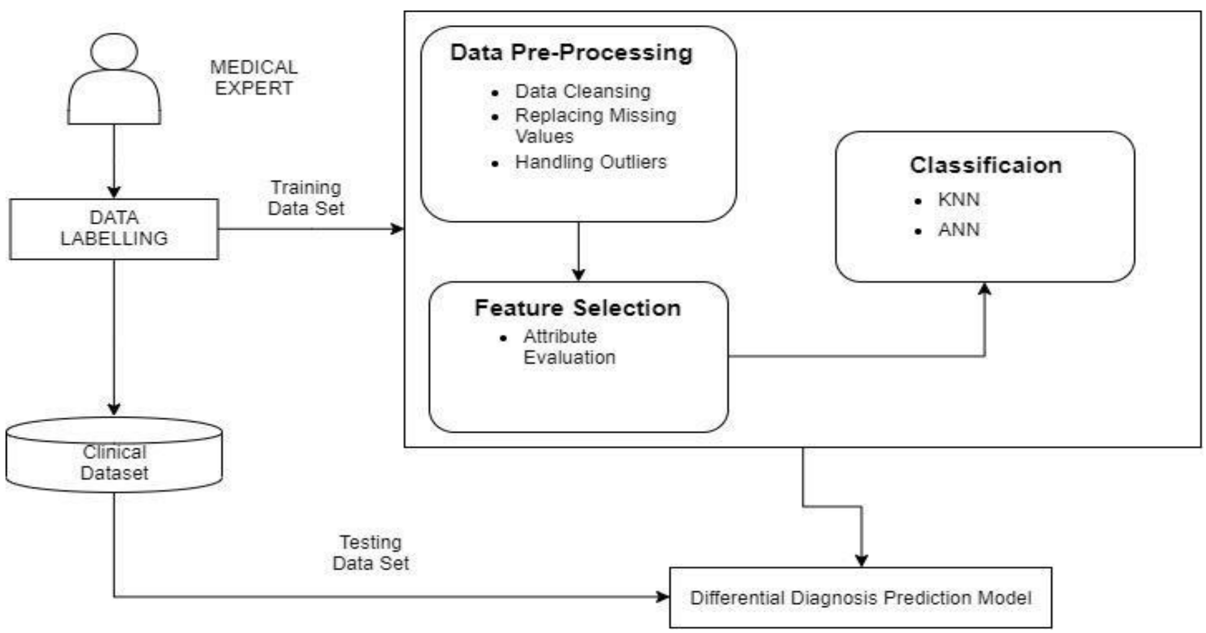


Figure 4.2: System Architecture

4.3 Data Flow Diagrams

4.3.1 Data Flow Diagram - Level 0

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

Level 0 describes the overall process of your project. It takes the input as disease data and processes through algorithm to get a trained model.

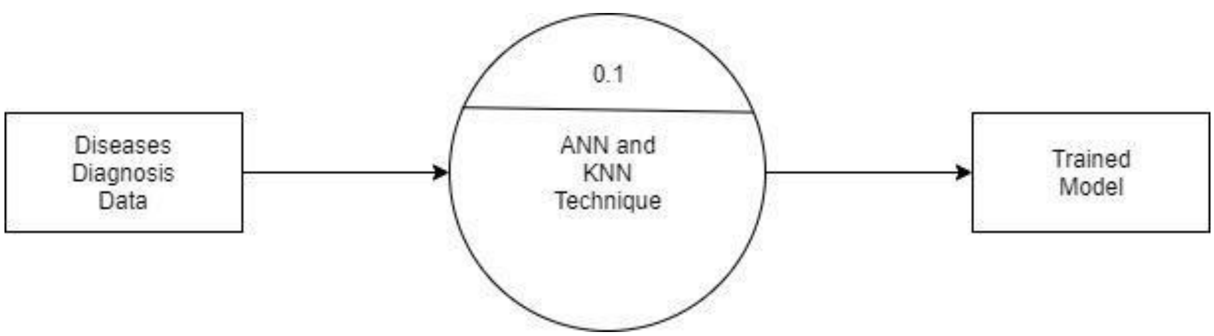


Figure 4.3.1: DFD-Level 0

4.3.2 Data Flow Diagram - Level 1

Level: 1 The below figure 4.3.2 describes the data flow of our project from one instance to the next. It takes the input of database and processes the data, performs the neural network algorithm to get a classified data.

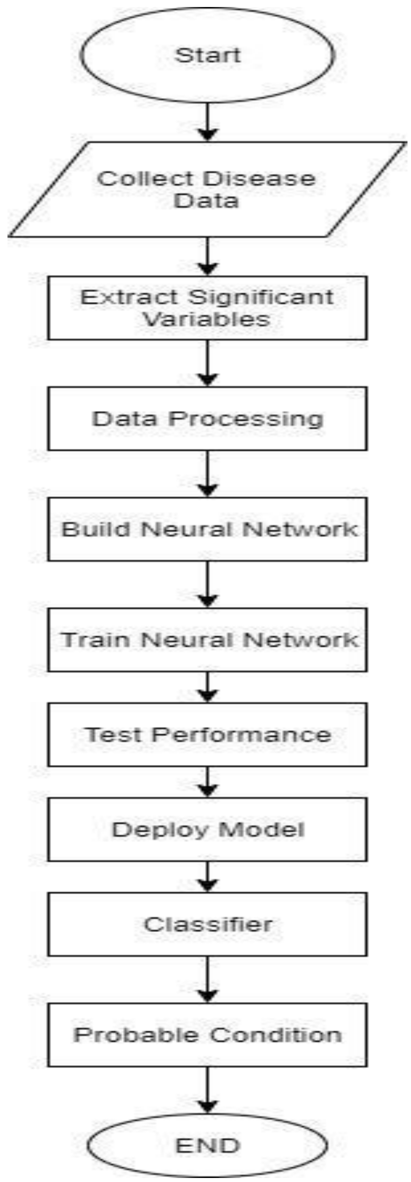


Figure 4.3.2: DFD-Level 1

4.3.3 Data Flow Diagram - Level 2

Level: 2 The below figure 4.3.3 describes the final step of our project. The classified data is used to take the inputs and give a probability of the disease occurring and the new data shall be updated in the database.

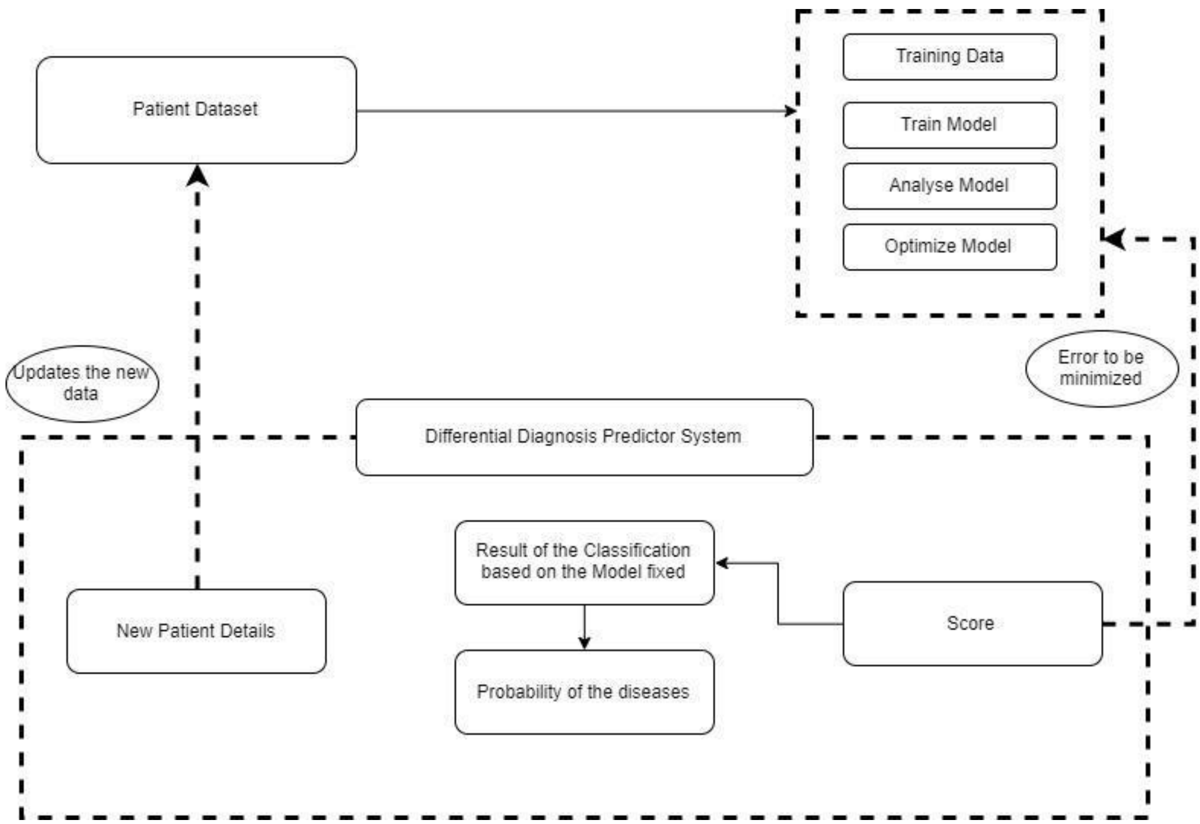


Figure 4.3.3: DFD-Level 2

4.4 Use Case Diagrams

The below figure 4.4 describes use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

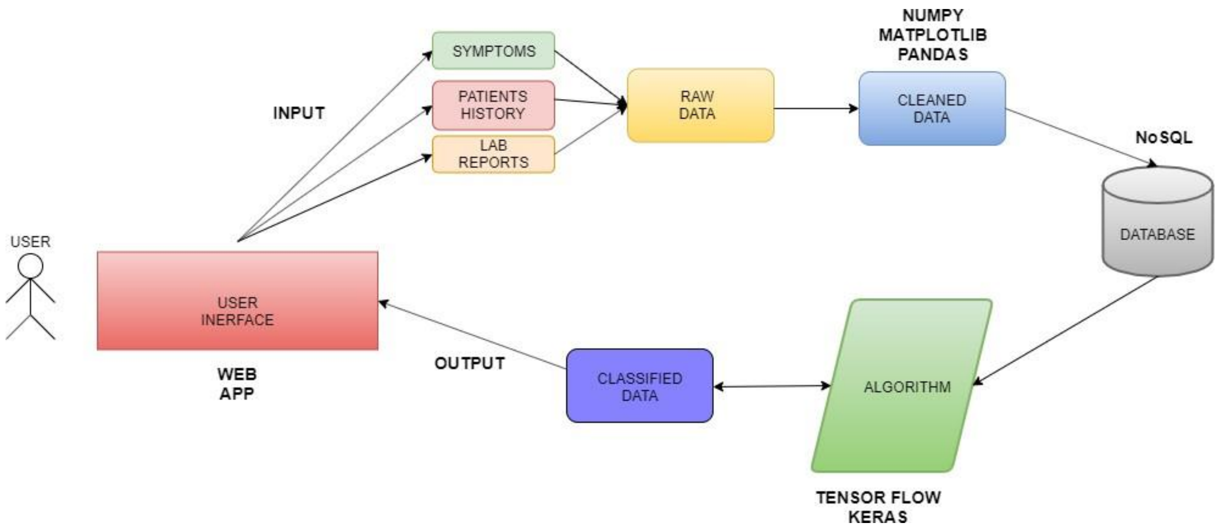


Figure 4.4: Use case diagram

4.5 Class Diagrams

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information. The below figure 4.5 describes the class diagram for the following system.

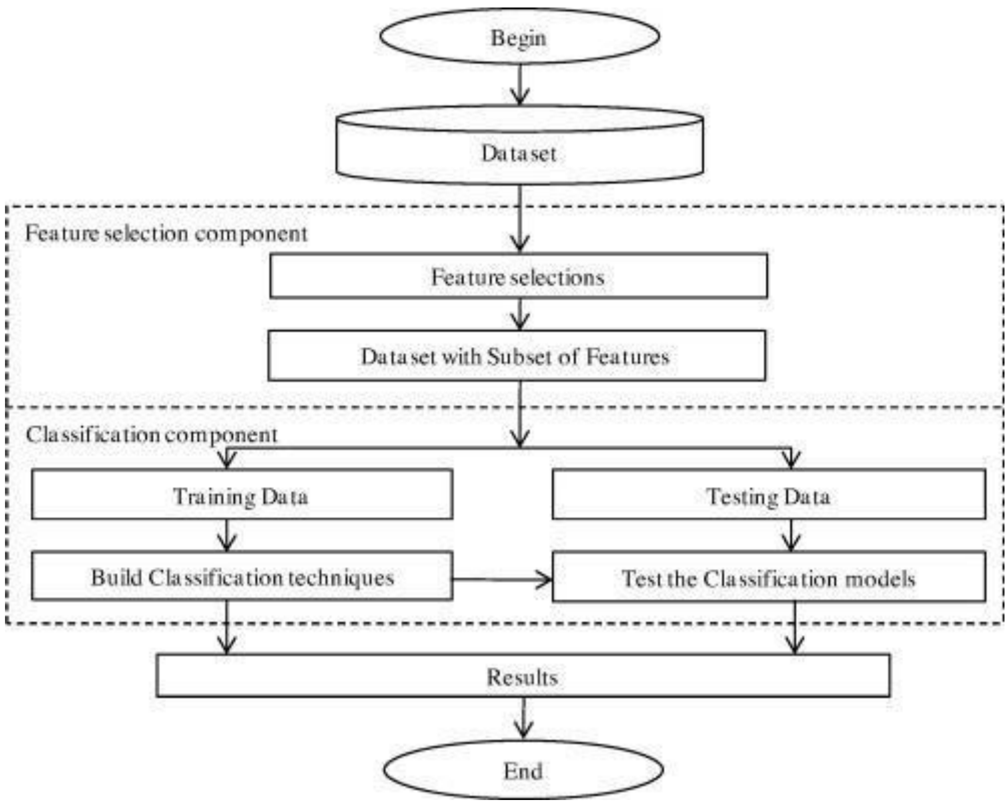


Figure 4.5: Class diagram

4.6 Sequence Diagrams

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. The below figure 4.6 describes the sequence diagram of the system.

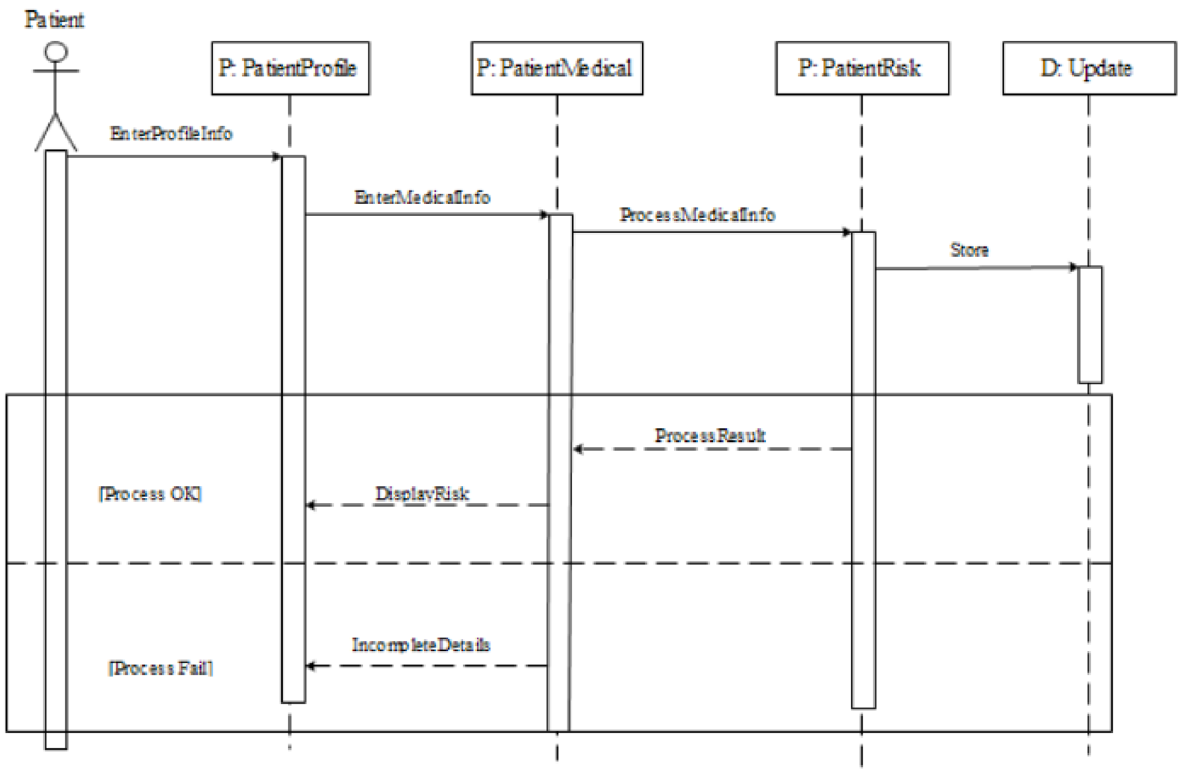


Figure 4.6: Sequence diagram

4.7 Activity Diagrams

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. The below figure 4.6 describes the activity diagram of the system.

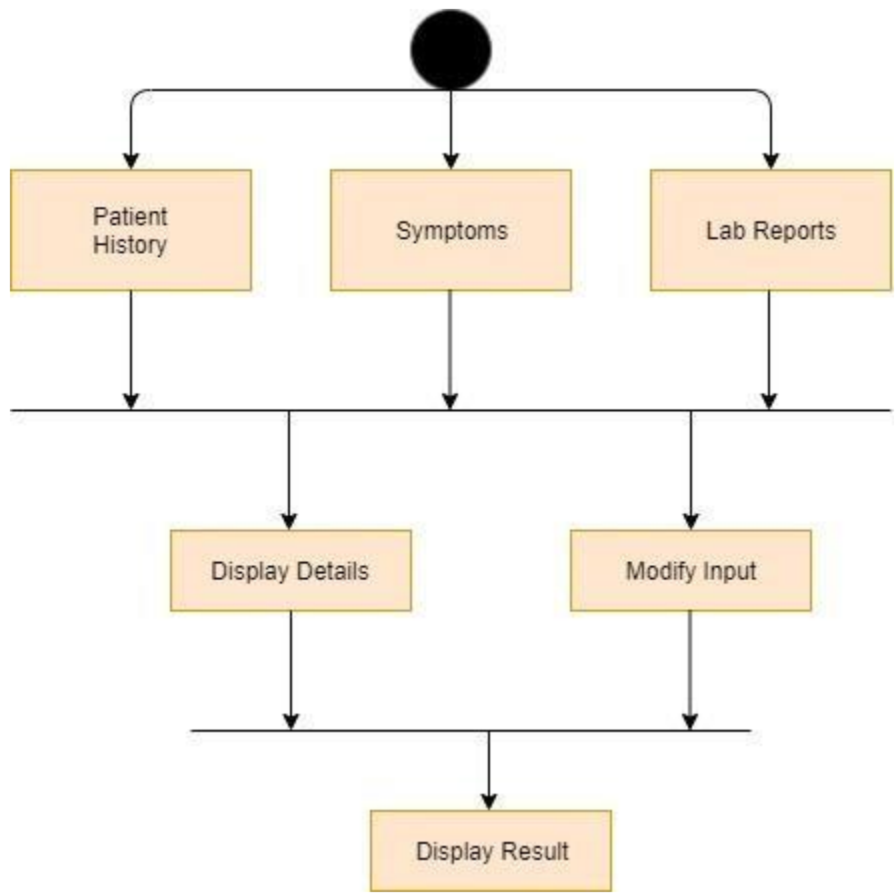


Figure 4.7: Activity diagram

4.8 Modules of the Project

4.9.1 Module 1

Module Name: - Data Pre-processing Phase

Functionality:-The function of data preprocessing is of collecting raw data and then transform this data into a format that can be processed by the python libraries.

Input:-Disease prediction database.

Output:-Processed dataset with attributes and values which are not dependent.

Libraries used: Python libraries namely numpy and pandas.

4.9.2 Module 2

Module Name:-Training the model.

Functionality:-The function of this phase is to train the model with the data which is available after processing using an algorithm.

Input:-Attributes of the dataset with their values.

Output:-The model after the learning process will try to predict the possibility of a disease from the given list of symptoms which are attributes in this case.

Algorithms Used:-KNN algorithm and ANN back-propagation algorithm.

4.9.3 Module 3

Module Name: - Output Phase.

Functionality:-In this module we will display the possible disease on the application designed by us.

Input: - The predicted disease with its probability.

Output: - Display of the disease with its probability and a brief description of the disease along with the possible symptoms which may further occur.

Chapter 5

IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed.

5.1 Steps for Implementation

The implementation is done on an interactive software simulation. A software simulation is a model of your software that allows you to demonstrate its key functions and operations. It helps is show the working and gives a closer look at the features it provides. Some of the software utilities used are:

Anaconda (Python distribution)

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

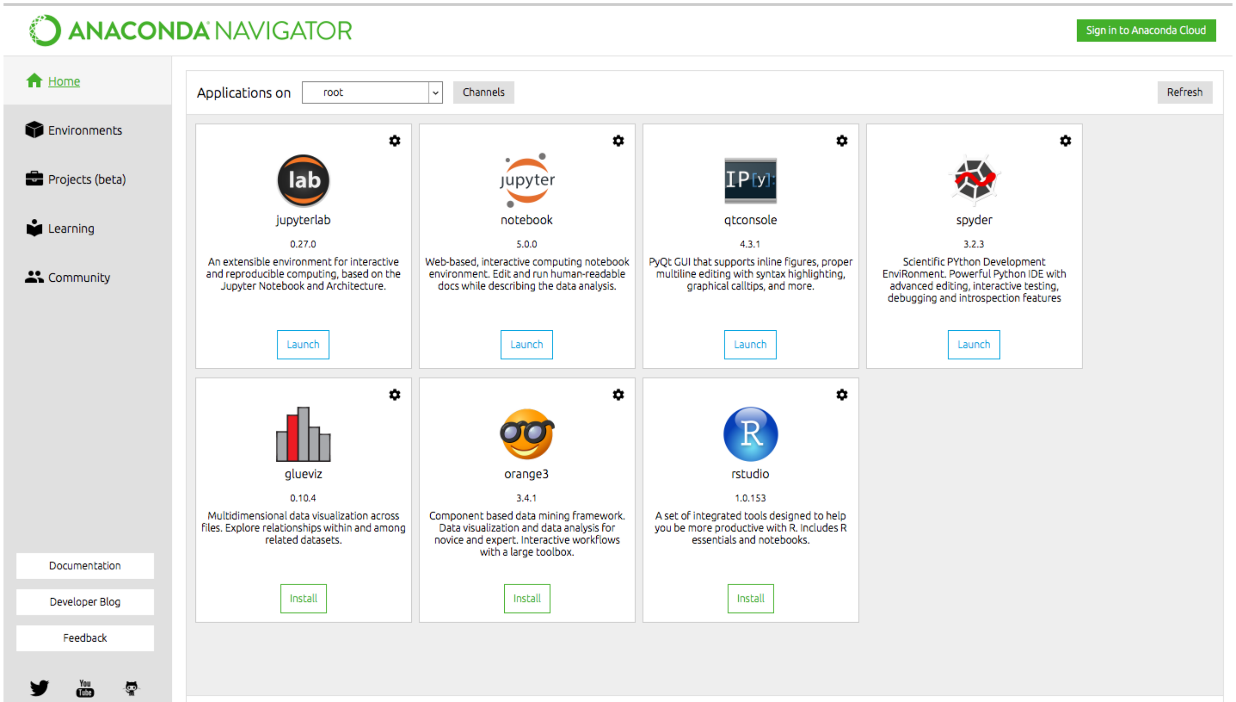


Figure 5.1: An overview of the Anaconda Navigator

The above figure 5.1 shows the user interface of anaconda navigator, and the use of Jupyter notebook for our system.

Jupyter Notebook

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension. A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell. To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user. Jupyter

Notebook can connect to many kernels to allow programming in many languages. By default Jupyter Notebook ships with the IPython kernel.

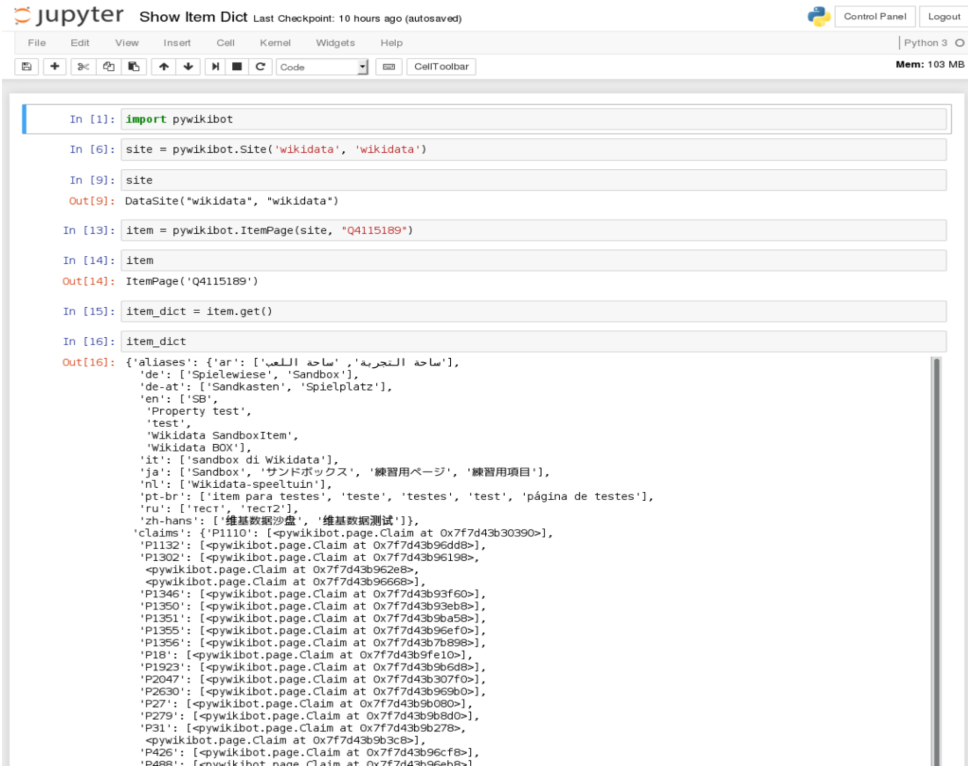


Figure 5.2: Jupyter Notebook Interface

The above figure 5.2 shows the user interface of Jupyter Notebook.

Django

Django is a free and open source web application framework written in Python. A framework is nothing more than a collection of modules that make development easier. They are grouped together, and allow you to create applications or websites from an existing source, instead of from scratch. Django offers a big collection of modules which you can use in your own projects. Primarily, frameworks exist to save developers a lot of wasted time and headaches and Django is no different. Django’s template language is designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed.

Sample Data

A dataset (or data set) is a collection of data, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. It lists values for each of the variables, such as height and weight of an object. Each value is known as a datum. The dataset may comprise data for one or more members, corresponding to the number of rows. Nontabular datasets can take the form of marked up strings of characters, such as an XML file.

We have looked up for open datasets that are present on Kaggle, UCI ML repositories and even looked up in case studies that are part of researches already concluded. These platoforms provide an abundance of data provided by companies and statisticians. This helps data miners and other researchers produce best models for predicting and describing data.

As key component for this project, datasets from a local diagnostic center is taken. The test results are from the blood tests done on a person having a liver disorder and not, and according to the diseases that we’ve taken. Each line in the dataset constitutes the record of a single individual of both the genders and varied range of age.

5.2 Implementation Issues

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified and so that debugging testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity clarity and elegance are the hallmarks of good programs and these characteristics have been implemented in each program module.

The goals of implementation are as follows.

- Minimize the memory required.
- Maximize output readability.
- Maximize source text readability.
- Minimize the number of source statements.
- Minimize development time.

5.3 Algorithms

5.3.1 Algorithm 1

KNN

1. Load the training and test data

2. Choose the value of K

3. For each point in test data:

- find the Euclidean distance to all training data points

- store the Euclidean distances in a list and sort it

- choose the first k points

- assign a class to the test point based on the majority of classes present in the chosen points

4. End

5.3.2 Algorithm 2

ANN

1. Generate the initial solution randomly Evaluate each individual in the population f(x) based on error rate.
2. Find the best solution from the population While (stopping criterion satisfied) For i 1 to n do Form = 1 to n do If (f(xj) < f(xi))
3. Calculate attractive fireflies by Eq.1
4. Calculate the distance between each fireflies i and j by Eq2
5. Move all firefly (r) to the best solution (x) by Eq3
6. End if, End for, j End for i. Moves best solution randomly by Eq. 4
7. Find the best solution from the new population End while Return best (TP) (NV) (FP), and (FV)
8. End of the algorithm

Chapter 6

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the 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.

6.1 Test Environment

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

6.2 Unit Testing Of Modules

6.2.1 Module 1

Data Description	Gender, Symptoms, Alcohol, Hepatitis B Surface Antigen, Hepatitis B e Antigen, Hepatitis B Core Antibody, Hepatitis C Virus Antibody, Cirrhosis, Endemic Countries, Smoking, Diabetes, Obesity, Hemochromatosis, Arterial Hypertension, Chronic Renal Insufficiency, Human Immunodeficiency Virus, Nonalcoholic Steatohepatitis, Esophageal Varices, Splenomegaly, Portal Hypertension, Portal Vein Thrombosis, Liver Metastasis, Radiological Hallmark, Age at diagnosis, Grams of Alcohol per day, Packs of cigarettes per year, Performance Status, Encefalopathy degree, Ascites degree, International Normalised Ratio, Alpha-Fetoprotein (ng/mL), Haemoglobin (g/dL),Mean Corpuscular Volume (fl),Leukocytes(G/L),Platelets (G/L),Albumin (mg/dL),Total Bilirubin(mg/dL),Alanine transaminase (U/L),Aspartate transaminase (U/L),Gamma glutamyl transferase (U/L),Alkaline phosphatase (U/L),Total Proteins (g/dL),Creatinine (mg/dL),Number of Nodules,Major dimension of nodule (cm),Direct Bilirubin (mg/dL),Iron (mcg/dL),Oxygen Saturation (%),Ferritin (ng/mL),Class
Test Data	(0,?,0,0,0,0,1,1,?,?,1,0,0,1,0,0,0,1,0,0,0,0,1,62,0,?,0,1,1,?,?,?,?,?,?,?,?,?,?,?,?,?,?,1,1.8,?,?,?,?,1)
Expected Results	(0,0,0,0,0,0,1,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,1,62,0,0,0,1,1,0.9,6,13,86,6,250000,4,0.6,31, 25,28,7,0.7,1,1.8,0.2,115,97.5,80,1)
Observed Results	(0,0,0,0,0,0,1,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,1,62,0,0,0,1,1,0.9,6,13,86,6,250000,4,0.6,31, 25,28,7,0.7,1,1.8,0.2,115,97.5,80,1)
Remarks	PASS

Table 6.1: Unit Test Case for Data Pre-processing

6.2.2 Module 2

Data Description	Gender,Symptoms,Alcohol,Hepatitis B Surface Antigen,Hepatitis B e Antigen,Hepatitis B Core Antibody,Hepatitis C Virus Antibody,Endemic Countries,Smoking,Diabetes,Obesity,Hemochromatosis,Arterial Hypertension,Chronic Renal Insufficiency,Human Immunodeficiency Virus,Nonalcoholic Steatohepatitis,Esophageal Varices,Splenomegaly,Portal Hypertension,Portal Vein Thrombosis,Liver Metastasis,Radiological Hallmark,Age at diagnosis,Grams of Alcohol per day, Packs of cigarettes per year
Test Data	(0,0,0,0,0,0,1,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,1,62,0,0,0,1,1,0.9,6,13,86,6,250000,4,0.6,31,25,28,7,0.7,1,1.8,0.2,115,97.5,80,1)
Expected Results	Class =1 (Cirrhosis positive)
Observed Results	Class =1 (Cirrhosis positive)
Remarks	PASS

Table 6.2: Test Case for Training the Model

6.3 Integration Testing Of Modules

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

1. Bottom-up Integration

This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

2. Top-down Integration

In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter.

In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations. Table 8.3.2 shows the test cases for integration testing and their results.

6.3.1 Module 1 and Module 2

Data Description	Gender,Symptoms,Alcohol,Hepatitis B Surface Antigen,Hepatitis B e Antigen,Hepatitis B Core Antibody,Hepatitis C Virus Antibody,Cirrhosis,Endemic Countries,Smoking,Diabetes,Obesity,Hemochromatosis,Arterial Hypertension,Chronic Renal Insufficiency,Human Immunodeficiency Virus,Nonalcoholic Steatohepatitis,Esophageal Varices,Splenomegaly,Portal Hypertension,Portal Vein Thrombosis,Liver Metastasis,Radiological Hallmark,Age at diagnosis,Grams of Alcohol per day,Packs of cigarets per year,Performance Status,Encefalopathy degree,Ascites degree,International Normalised Ratio,Alpha-Fetoprotein (ng/mL),Haemoglobin (g/dL),Mean Corpuscular Volume (fl),Leukocytes(G/L),Platelets (G/L),Albumin (mg/dL),Total Bilirubin(mg/dL),Alanine transaminase (U/L),Aspartate transaminase (U/L),Gamma glutamyl transferase (U/L),Alkaline phosphatase (U/L),Total Proteins

	(g/dL),Creatinine (mg/dL),Number of Nodules, Major dimension of nodule (cm),Direct Bilirubin (mg/dL),Iron (mcg/dL),Oxygen Saturation (%),Ferritin (ng/mL),Class
Test Data	(0,0,0,0,0,0,1,1,0,0,1,0,0,1,0,0,0,1,0,0,0,0,1,62,0,0,0,1,1,0.9,6,13,86,6,250000,4,0.6,31, 25,28,7,0.7,1,1.8,0.2,115,97.5,80,1)
Expected Results	Class =1 (Cirrhosis positive)
Observed Results	Class =1 (Cirrhosis positive)
Remarks	PASS

Table 6.3: Integration Testing for Module 1 and Module 2

6.4 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is important because of the following reasons:

- System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

- The application is tested thoroughly to verify that it meets the functional and technical specifications.

- The application is tested in an environment that is very close to the production environment where the application will be deployed.

- System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

Data Description	Gender,Symptoms,Alcohol,Hepatitis B Surface Antigen,Hepatitis B e Antigen,Hepatitis B Core Antibody,Hepatitis C Virus Antibody,Cirrhosis,Endemic Countries,Smoking,Diabetes,Obesity,Hemochromatosis,Arterial Hypertension,Chronic Renal Insufficiency,Human Immunodeficiency Virus,Nonalcoholic Steatohepatitis,Esophageal Varices,Splenomegaly,Portal Hypertension,Portal Vein Thrombosis,Liver Metastasis,Radiological Hallmark,Age at diagnosis,Grams of Alcohol per day,Packs of cigarets per year,Performance Status,Encefalopathy degree,Ascites degree,International Normalised Ratio,Alpha-Fetoprotein (ng/mL),Haemoglobin (g/dL),Mean Corpuscular Volume (fl),Leukocytes(G/L),Platelets (G/L),Albumin (mg/dL),Total Bilirubin(mg/dL),Alanine transaminase (U/L),Aspartate transaminase (U/L),Gamma glutamyl transferase (U/L),Alkaline phosphatase (U/L),Total Proteins (g/dL),Creatinine (mg/dL),Number of Nodules,Major dimension of nodule (cm),Direct Bilirubin (mg/dL),Iron (mcg/dL),Oxygen Saturation (%),Ferritin (ng/mL),Class
Test Data	(1,1,1,1,0,1,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,1,76,100, 30,12,1,1,1.56,50655,9.8,85.6,3900,132000,2.6,2.6,123,219,503,363,7.3,0.55,1,4,1.5,40,12,57,1)
Expected Results	Class =1 (Cirrhosis positive)
Observed Results	Cirrhosis
Remarks	PASS

Table 6.4: System Testing for Cirrhosis

Data Description	Gender,Symptoms,Alcohol,Hepatitis B Surface Antigen,Hepatitis B e Antigen,Hepatitis B Core Antibody,Hepatitis C Virus Antibody,Cirrhosis,Endemic Countries,Smoking,Diabetes,Obesity,Hemochromatosis,Arterial Hypertension,Chronic Renal Insufficiency,Human Immunodeficiency Virus,Nonalcoholic Steatohepatitis,Esophageal Varices,Splenomegaly,Portal Hypertension,Portal Vein Thrombosis,Liver Metastasis,Radiological Hallmark,Age at diagnosis,Grams of Alcohol per day,Packs of cigarets per year,Performance Status,Encefalopathy degree,Ascites degree,International Normalised Ratio,Alpha-Fetoprotein (ng/mL),Haemoglobin (g/dL),Mean Corpuscular Volume (fl),Leukocytes(G/L),Platelets (G/L),Albumin (mg/dL),Total Bilirubin(mg/dL),Alanine transaminase (U/L),Aspartate transaminase (U/L),Gamma glutamyl transferase (U/L),Alkaline phosphatase (U/L),Total Proteins (g/dL),Creatinine (mg/dL),Number of Nodules,Major dimension of nodule (cm),Direct Bilirubin (mg/dL),Iron (mcg/dL),Oxygen Saturation (%),Ferritin (ng/mL),Class
Test Data	(1,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,0,1,61,60,67.5,2,0,1,1,1.13 75,13.3,90,8,385000,4.3 0.6,53,52,164,181,7.5,1.46,5,18.6,0.2,115,97.5,80)
Expected Results	Class =1 (HCC positive)
Observed Results	HCC
Remarks	PASS

Table 6.5: System Testing for Hcc

Data Description	Gender,Symptoms,Alcohol,Hepatitis B Surface Antigen,Hepatitis B e Antigen,Hepatitis B Core Antibody,Hepatitis C Virus Antibody,Cirrhosis,Endemic Countries,Smoking,Diabetes,Obesity,Hemochromatosis,Arterial Hypertension,Chronic Renal Insufficiency,Human Immunodeficiency Virus,Nonalcoholic Steatohepatitis,Esophageal Varices,Splenomegaly,Portal Hypertension,Portal Vein Thrombosis,Liver Metastasis,Radiological Hallmark,Age at diagnosis,Grams of Alcohol per day,Packs of cigarets per year,Performance Status,Encefalopathy degree,Ascites degree,International Normalised Ratio,Alpha-Fetoprotein (ng/mL),Haemoglobin (g/dL),Mean Corpuscular

	Volume (fl),Leukocytes(G/L),Platelets (G/L),Albumin (mg/dL),Total Bilirubin(mg/dL),Alanine transaminase (U/L),Aspartate transaminase (U/L),Gamma glutamyl transferase (U/L),Alkaline phosphatase (U/L),Total Proteins (g/dL),Creatinine (mg/dL),Number of Nodules,Major dimension of nodule (cm),Direct Bilirubin (mg/dL),Iron (mcg/dL),Oxygen Saturation (%),Ferritin (ng/mL),Class
Test Data	(0,1,1,1.79,44340,12.7,95.8,14.4,4,12000,2.2,28.5,78,127,444,462,6.6,3.955,8.5,19.8,115, 97.5,80,0,0,1,1,1.79,44340,12.7,95.8,14.44,12000,2.2,28.5,78,127444,462,6.6,3.95,5,8.5,19.8, 115,97.5,80,0)
Expected Results	Class =1 (Cirrhosis positive)
Observed Results	Normal
Remarks	PASS

Table 6.6: System Testing for Normal

Chapter 7

RESULTS

This section describes the screens of the Expert Systems for Differential Diagnosis Using Deep Learning. The snapshots are shown below for each module.

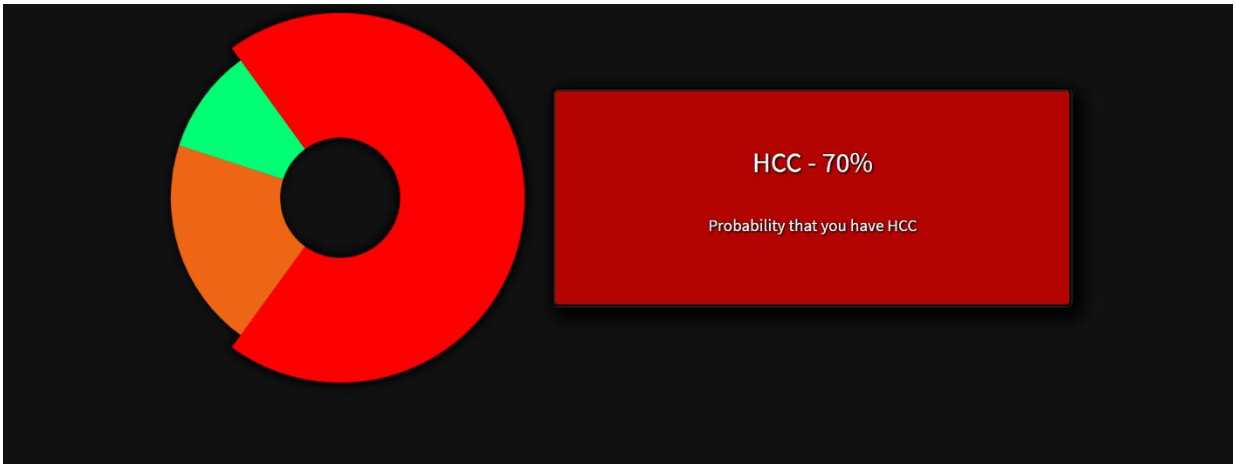
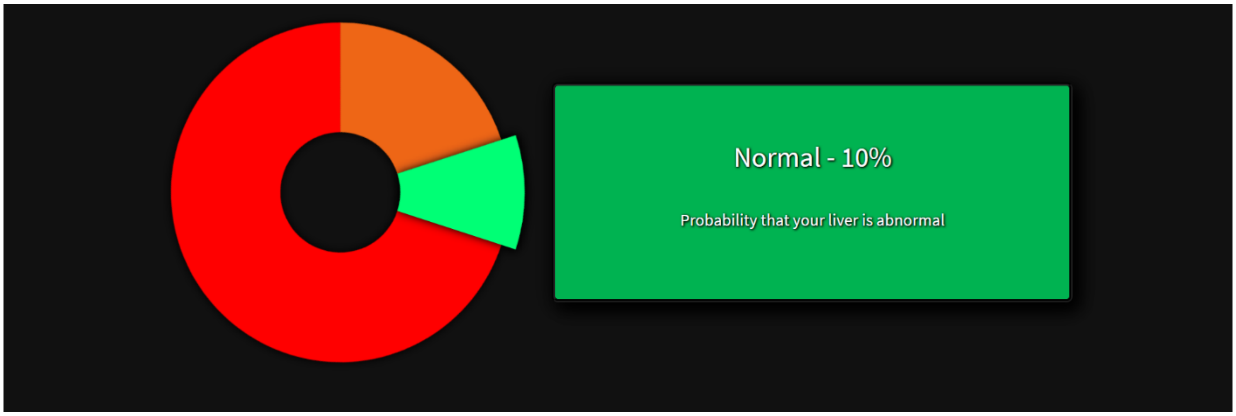


Figure 7.1: Prediction of HCC

The above figure 7.1 shows us the result for a particular dataset which gives a 70% probability of HCC liver disease. It's to be noted that this is the closest prediction of the system model.

Figure 7.2: Prediction of Cirrhosis

The above figure 7.2 shows the result for the same dataset that was previously used and gives us a 20% probability of Cirrhosis liver disease. We can note that there is also a possibility of both the disease being present but however the other disease has a more probability.



Snapshot 7.3: Prediction of Normality

The above figure 7.3 shows the result for the same data dataset that was previously used and gives a 10% chance of normality which is lesser than the possibility of the disease being present.

CONCLUSION

In a time when healthcare is becoming the most important sector of the society, reducing the time of detection and prioritizing treatment becomes utmost important. Trial to do so with the help of latest technology such as differential diagnosis with the knowledge of neural network has been carried out in this project. This expert system will take the symptoms as input and with the use of modulation of data and algorithms gives us the closest prediction of liver diseases. This simulation model uses latest technology and gives the user an interface that's user-friendly to make use of. Not only is it helpful but also accessible to people of all age due to its accessibility. With this, we have tried make full use of the most critical golden time of the patient.

8.1 Future Enhancements

In future more ML models can be implemented to make it more convenient for using it. Also include more organs for diagnosing and giving it vast scope.

Some of the feature enhancements that are intended to implement are:

- Use image recognition for scanning the reports and filling details automatically.
- Include more organs and help doctors in diagnosing highly complicated cases.
- Provide suggestions on treatment and advices from expertise.
- Prescribe appropriate medicines.

BIBLIOGRAPHY

[1] Murali Ravuri, Anitha Kannan, “From expert system to machine-learned diagnosis models, Proceedings of Machine Learning Research , Machine Learning for Healthcare,85,1–16, 2018 IEEE.

[2] Dhaval Raval, Dvijesh Bhatt, “Medical diagnosis system using Machine Learning, International Journal of computer science , Machine Learning for Healthcare, Volume 7,1–6, 2016 IEEE.

[3]Dr A N Jayanthi, Ms D Pavithra, “A Study on Machine Learning Algorithm in Medical Diagnosis”, International Journal of Advanced Research in Computer Science, Volume 9, No 4, July-August 2018 IEEE.

[4] Shaik Razia,N Vamsi Krishna,”A Review On Disease Diagnosis Using Machine Learning Techniques” , International Journal of Pure and Applied Mathematics, Volume 117 No. 16 2017 IEEE, 79-85.

[5] Alex Yee,” Machine Learning Applied to the Detection of Retinal Blood Vessels, Stanford University.