A

Major Project

On

HEART DISEASE PREDICTION USING BIO INSPIRED ALGORITHMS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

By

KASARLA ASHWITHA (207R1A6787)
VULLENGA NAREN RAO (207R1A67C0)
MANE SAI GANESH (207R1A6795)

Under the Guidance of

Ms.SANDHYA RANI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE.NewDelhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401. **2020-2024**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled **"Heart Disease Prediction Using Bio Inspired Algorithms"** being submitted by **KASARLA ASHWITHA (207R1A6787), VULLENGA NAREN RAO (207R1A67C0) & MANE SAI GANESH (207R1A6795)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (DATA SCIENCE) to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Ms.SANDHYA RANI	**Dr. K. SRINIVAS**
Assistant Professor	
INTERNAL GUIDE	HOD CSE (DATA SCIENCE)

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ___________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Ms.Sandhya Rani,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. B. Shankar Nayak,Ms.Sandhya Rani & Mr.M.Praveen** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srinivas ,** Head, Department of Computer Science and Engineering (DATA SCIENCE) for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the courseof this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

KASARLA ASHWITHA	**(207R1A6787)**
VULLENGA NAREN RAO	**(207R1A67C0)**
MANE SAI GANESH	**(207R1A6795)**

ABSTRACT

Heart disease is a major health concern worldwide, and early prediction can significantly improve patient outcomes. In this project, we explore the use of bio-inspired algorithms, which mimic natural processes, to predict heart disease risk. These algorithms, such as genetic algorithms and particle swarm optimization, are modeled after biological phenomena like evolution and collective behavior of organisms. By applying these algorithms to analyze medical data including patient demographics, lifestyle factors, and medical history, we aim to develop a robust prediction model. This model will assist healthcare professionals in identifying individuals at high risk of heart disease, enabling timely interventions and personalized treatment plans. Our approach offers a novel and effective method for heart disease prediction, contributing to improved healthcare management and patient well-being.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROJECT SCOPE

The project scope involves using bio-inspired algorithms, which are computer programs inspired by natural processes, to predict the risk of heart disease. We'll collect and analyze various types of medical data from patients, such as age, gender, lifestyle habits, and medical history. By applying these algorithms to this data, we aim to create a reliable system that can identify individuals who are at high risk of developing heart disease.

1.2 PROJECT PURPOSE

The purpose of the project is to develop a tool that can predict the likelihood of someone developing heart disease using computer programs inspired by nature. By collecting information about a person's age, lifestyle, and medical history, we'll use these programs to analyze the data and identify individuals who might be at a higher risk of heart problems. This will help doctors intervene early, providing better care and potentially preventing heart disease before it becomes serious.

1.3 PROJECT FEATURES

The project features include the utilization of advanced computer algorithms inspired by nature to predict heart disease risk. The key features aim to provide accurate, accessible, and up-to-date predictions to support proactive healthcare interventions and improve patient outcomes..

2.SYSTEM ANALYSIS

SYSTEM ANALYSIS

The system analysis of the project involves thoroughly examining how the heart disease prediction tool functions and interacts with various components. This includes understanding the input data sources such as patient demographics and medical records, as well as the algorithms used to process this information. Additionally, the analysis assesses the user interface design to ensure it's user-friendly for healthcare professionals.

2.1 PROJECT DEFINITION

The project, "Heart Disease Prediction Using Bio Inspired Algorithms," aims to develop a predictive system that assesses the risk of heart disease in individuals by employing computational algorithms inspired by natural processes. This project entails collecting relevant medical data, applying bio-inspired algorithms to analyze the data, and generating predictions to aid in early detection and prevention of heart-related conditions.

2.2 EXISTING SYSTEM

Heart diseases have emerged as one of the most prominent cause of death all around the world. According to World Health Organisation, heart related diseases are responsible for the taking 17.7 million lives every year, 31% of all global deaths. In India too, heart related diseases have become the leading cause of mortality [1]. Heart diseases have killed 1.7 million Indians in 2016, according to the 2016 Global Burden of Disease Report, released on September 15,2017. Heart related diseases increase the spending on health care and also reduce the productivity of an individual. Estimates made by the World Health Organisation (WHO), suggest that India have lost up to $237 billion, from 2005-2015, due to heart related or Cardiovascular diseases.Thus, feasible and accurate prediction of heart related diseases is very important. Medical organisations, all around the world, collect data on various health related issues. These data can be using various machine learning techniques to gain useful insights. But the data collected is very massive and, many a times, this data can be very noisy. Thus, these algorithms have become very useful, in recent times, to predict the presence or absence of heart related diseases accurately.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- Hard to Explain
- Needs Lots of Good Data
- Uses a Lot of Computer Power
- Health Information Changes
- Costs a Lot
- Doctors and Computers Need to Get Along

2.3 PROPOSED SYSTEM

Dimensionality Reduction involves selecting amathematical representation such that one can relate the majority of, but not all, the variance within the given data, thereby including only most significant information. The data considered for a task or a problem, may consists of a lot of attributesor dimensions, but not all of these attributes may equally influence the output. A large number of attributes, or features, may affect the computational complexity and may even lead to overfitting which leads to poor results. Thus, Dimensionality Reduction is a very important step considered while building any model. Dimensionality Reduction is generally achieved by two methods -Feature Extraction and Feature Selection.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Early Detection
- Accurate Prediction
- Continuous Improvement
- Cost-Effective
- Fair and Inclusive
- Complementary Support for Healthcare Professionals
- Adaptability to Changing Conditions
- Optimized Feature Selection

2.4 FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

2.4.1 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

2.4.2 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain

2.4.3 TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

• PROCESSOR	: intel core i3 or above
• RAM	: 4GB (min)
• HARD DISK	: 500 GB
• KEYBOARD	: Standard Windows Keyboard
• MOUSE	: Two or Three Button Mouse
• MONITOR	: SVGA

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

• OPERATING SYSTEM	: Windows 10 and above
• CODE LANGUAGE	: Python
• FRONT-END	: Python
• BACK-END	: Django-ORM
• DESIGNING	: HTML, CSS, JavaScript
• DATABASE	: MySQL (WAMP Server)

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

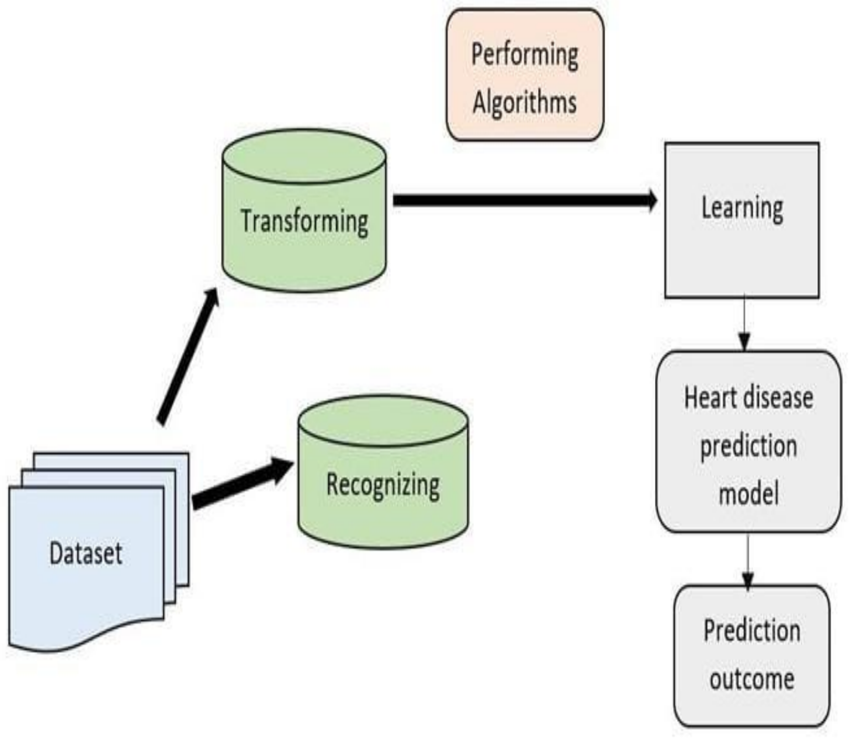


Figure 3.1: Project Architecture of Heart Disease Prediction Using Bio Inspired Algorithms.

3.2 DESCRIPTION

The project aims to contribute to the improvement of mental health detection practices, particularly in the context of Anorexia, by combining technological innovation with ethical considerations, user-centric design, and a commitment to advancing mental health research.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

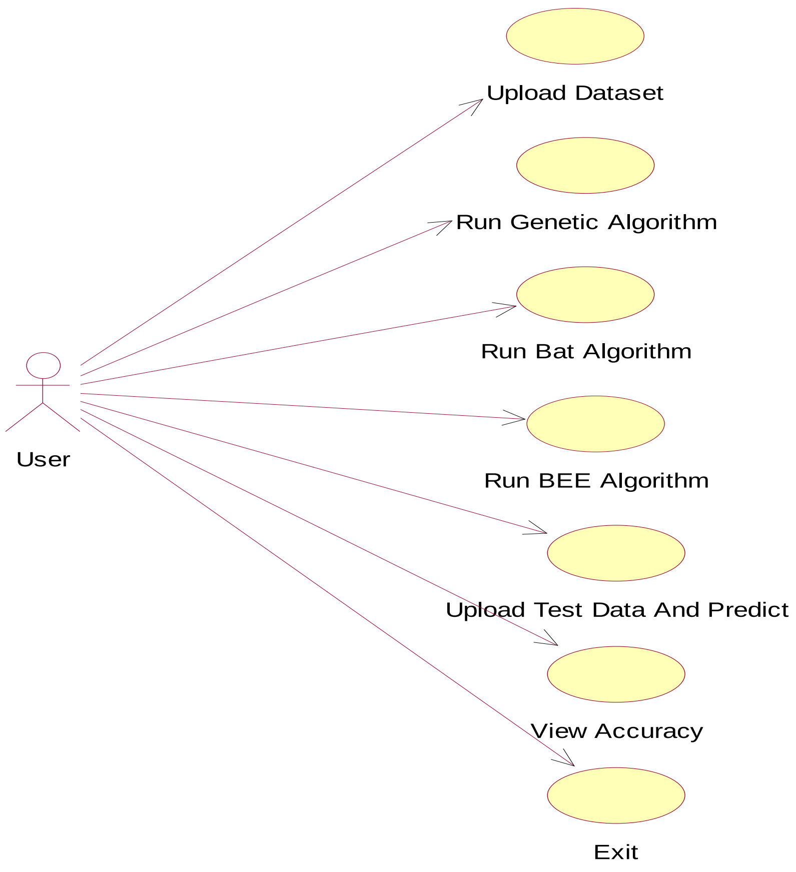


Figure 3.3: Use case diagram of Heart Disease Prediction Using Bio Inspired Algorithms.

3.4 CLASS DIAGRAM

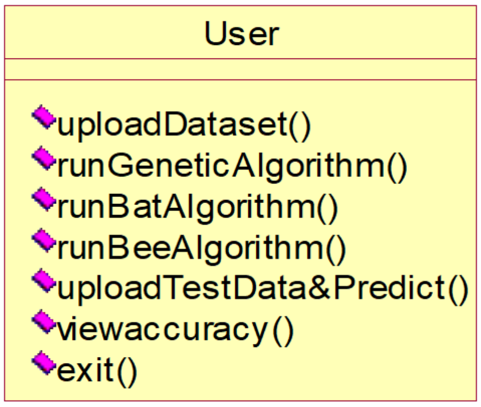


Figure 3.4: Class diagram of Heart Disease Prediction Using Bio Inspired Algorithms.

Class diagram 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 objects.

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

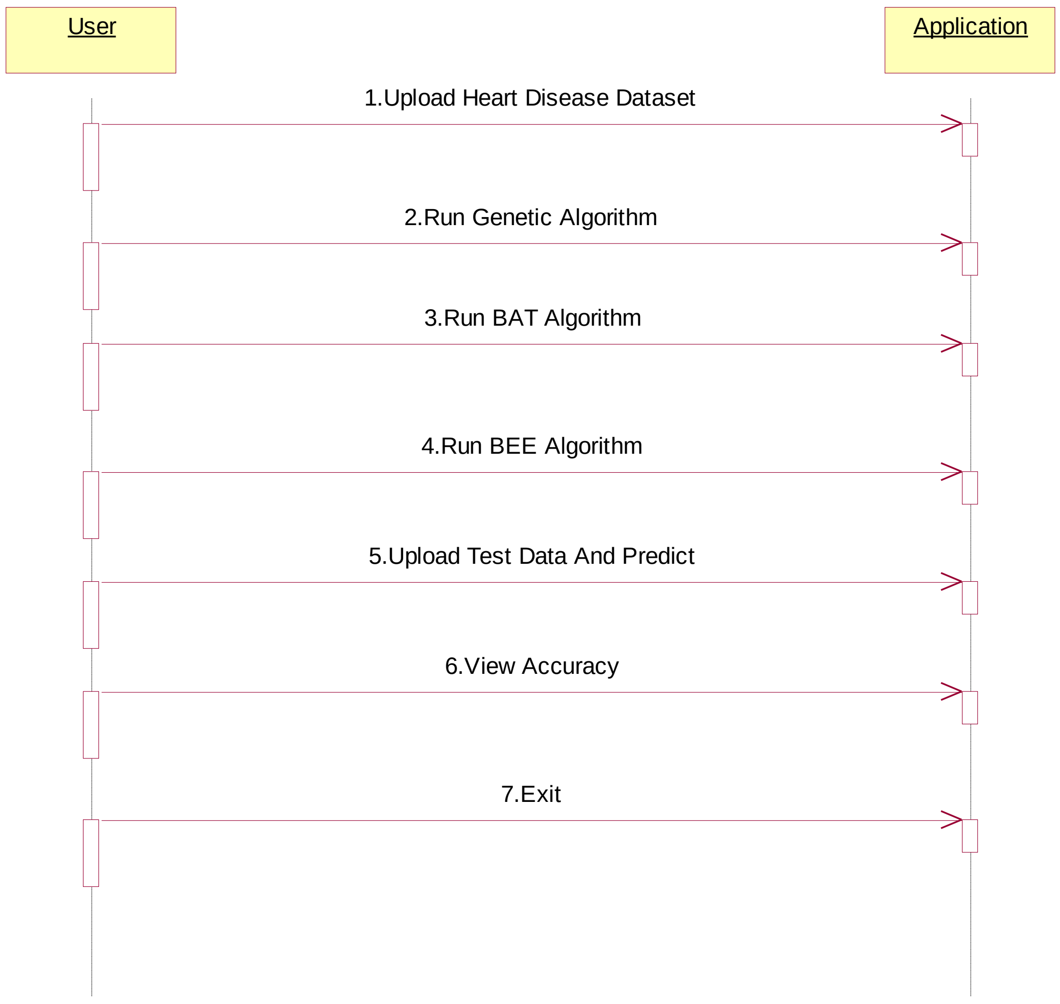


Figure 3.5: Sequence Diagram of Heart Disease Prediction Using Bio Inspired Algorithms.

3.6 COLLOBORATION DIAGRAM

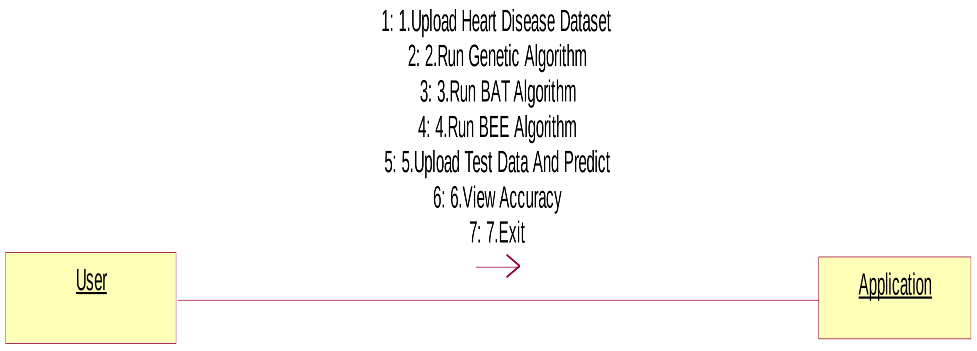

1: 1.Upload Heart Disease Dataset
2: 2.Run Genetic Algorithm
3: 3.Run BAT Algorithm
4: 4.Run BEE Algorithm
5: 5.Upload Test Data And Predict
6: 6.View Accuracy
7: 7.Exit
User
Application

Figure 3.6: Colloboration diagram of Heart Disease Prediction Using Bio Inspired Algorithms.

4.IMPLEMENTATION

4.1 RESEARCH METHODS

4.1.1 RANDOM FOREST ALGORITHM

Random Forest is one of the most popular and commonly used algorithms by Data Scientists. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks.

Steps Involved in Random Forest Algorithm:

Step 1: In the Random Forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression, respectively

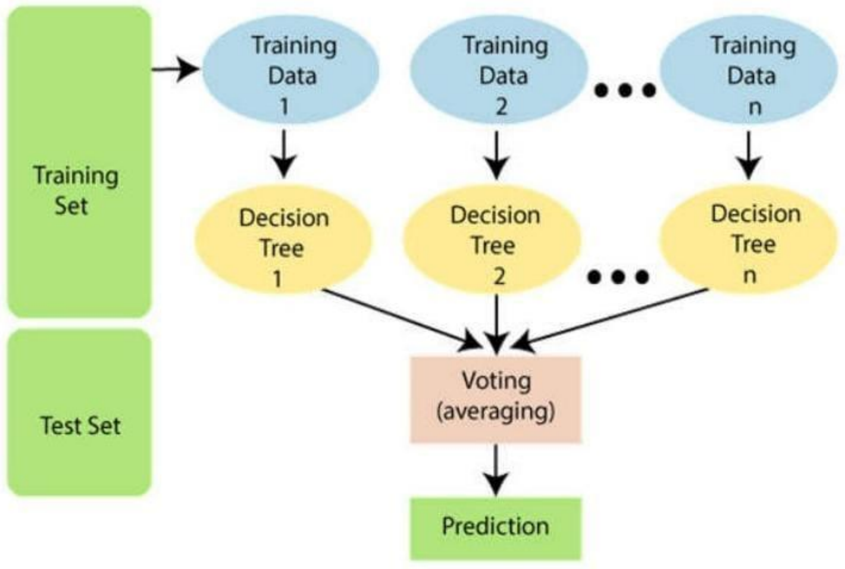


Figure 4.1.1:Random Forest Algorithm

4.1.2 GENETIC SELECTION CV

A genetic algorithm is a technique for optimization problems based on natural selection. The initial population (of size ‘n_population’) is generated at random from the sample space of feature sets. These sets are limited in scope by the parameter ‘max_features’, which sets the maximum size of each feature subset. For each member of the initial population, a score is measured with the target metric. This measurement is the performance of the estimator specified. A tournament selection is performed to determine which members will continue to the next generation. The number of members within the tournament is set with ‘tournament_size’. Tournament size is a selection of a few members from the population that compete against one another based on the scoring metric. The winner of a tournament is chosen as a parent for the next generation.

4.1.3 BAT ALGORITHM

The Bat Algorithm (BA) is a metaheuristic optimization algorithm inspired by the behavior of bats in nature.

1. Initialization: Generate a population of bats randomly.
2. Evaluation: Evaluate the objective function for each bat.
3. Update the best solution: Track the bat with the best solution.
4. Update positions and frequencies: Update bat positions based on the best solution and random exploration. Adjust bat frequencies.
5. Update velocities: Update bat velocities based on position changes.
6. Apply boundaries: Ensure bat positions stay within the search space.
7. Check convergence: Stop if the stopping criteria are met. Otherwise, go to step 2.
8. Return the best solution found.

The Bat Algorithm mimics bat echolocation behavior to explore and converge towards the optimal solution. It adjusts positions, frequencies, and velocities to navigate the search space. It is applicable to various machine learning optimization tasks.

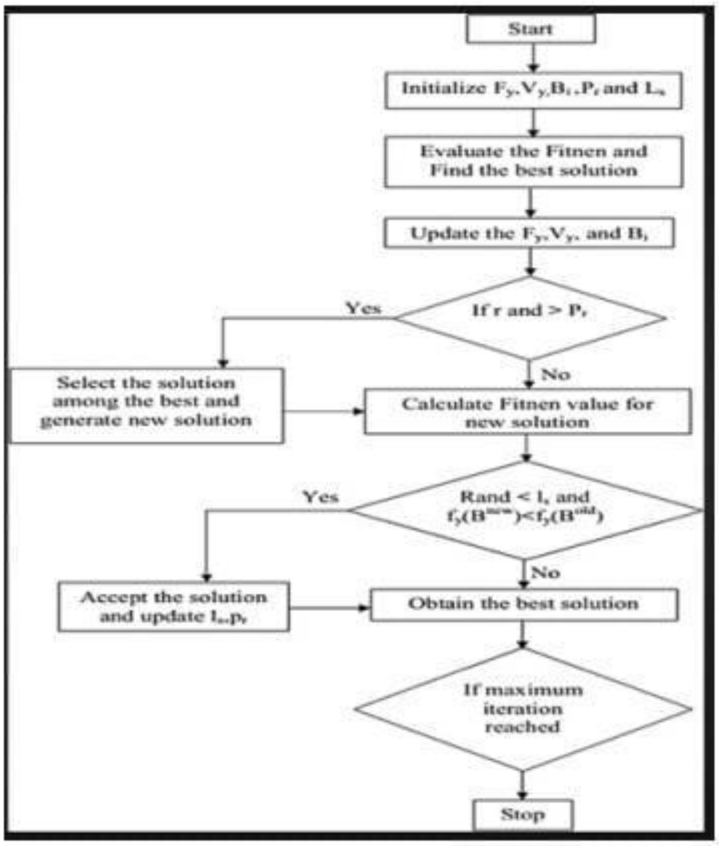

Start
Initialize F_y, V_y, B_i, P_r and L_s
Evaluate the Fitnen and Find the best solution
Update the F_y, V_y, and B_i
If r and > P_r
Yes
No
Select the solution among the best and generate new solution
Calculate Fitnen value for new solution
Rand < l_s and f_y(B^new)<f_y(B^old)
Yes
No
Accept the solution and update l_s, p_r
Obtain the best solution
If maximum iteration reached
Stop

Figure 4.1.3: BAT ALGORITHM

4.1.4 BEE ALGORITHM

The Artificial Bee Colony (ABC) algorithm is a population-based optimization algorithm inspired by the foraging behavior of honey bees. The ABC algorithm is commonly used in solving various optimization problems in machine learning.

Here's a concise explanation of steps included:

1. Initialization:

Generate an initial population of artificial bees randomly.

2. Employed bees phase:

- Each employed bee explores a solution in its neighborhood.
- Evaluate the objective function for each solution and determine its fitness.

3. Onlooker bees phase:

- Onlooker bees select solutions probabilistically based on their fitness.

- Fit solutions have a higher chance of being selected.

4. Update solution:

- Employed and onlooker bees generate new solutions through local search.
- The best solution is updated if a newly generated solution is better.

5. Scout bees phase:

- If a bee's solution remains unchanged for a certain number of iterations, it becomes a scout.
- The scout bee abandons its solution and generates a new random solution.

6. Check convergence:

- If the stopping criteria (maximum number of iterations or desired accuracy) are met, stop the algorithm. Otherwise, go to step 2.

7. Return the Best solution Found.

The ABC algorithm models the foraging behavior of honey bees, where employed bees explore the environment for nectar sources and onlooker bees select profitable sources based on the waggle dance communication. By employing local search and introducing scout bees, the algorithm explores the search space and converges towards the optimal solution.

4.2 SAMPLE CODE

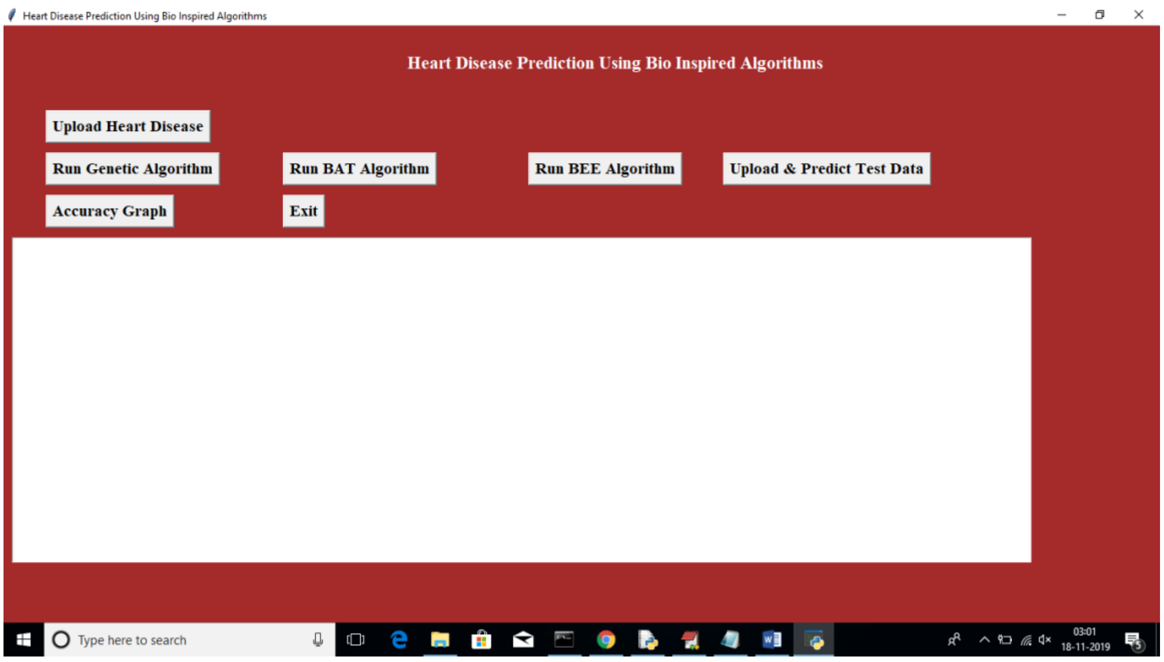
```
from __future__ import print_function
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
from tkinter.filedialog import askopenfilename
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import os
import re
from sklearn.metrics import accuracy_score
import numpy as np
from sklearn import datasets, linear_model
import pandas as pd
from genetic_selection import GeneticSelectionCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import SwarmPackagePy
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from BAT import BAT
from SwarmPackagePy import testFunctions as tf
from BEE import BEE
main = tkinter.Tk()
main.title("Heart Disease Prediction Using Bio Inspired Algorithms")
main.geometry("1300x1200")
global filename
global train
global ga_acc, bat_acc, bee_acc
global classifier
def upload():
    global filename
    filename = filedialog.askopenfilename(initialdir="heart_dataset")
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");
 def prediction(X_test, cls):  #prediction done here
    y_pred = cls.predict(X_test)
    for i in range(len(X_test)):
      print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
    return y_pred
def cal_accuracy(y_test, y_pred, details):
    cm = confusion_matrix(y_test, y_pred)
    accuracy = accuracy_score(y_test,y_pred)*100
    text.insert(END,details+"\n\n")
    text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")
    text.insert(END,"Report : "+str(classification_report(y_test, y_pred))+"\n")
    text.insert(END,"Confusion Matrix : "+str(cm)+"\n\n\n\n\n")
    return accuracy
def geneticAlgorithm():
    global classifier
    text.delete('1.0', END)
    global ga_acc
    train = pd.read_csv(filename)
    test = pd.read_csv('heart_dataset/test.txt')
    test_X = test.values[:, 0:12]
    X = train.values[:, 0:12]
    y = train.values[:, 13]
estimator = linear_model.LogisticRegression(solver="liblinear", multi_class="ovr")
selector = GeneticSelectionCV(estimator,
                              cv=5,
                              verbose=1,
                              scoring="accuracy",
                              max_features=10,
                              n_population=50,
                              crossover_proba=0.5,
                              mutation_proba=0.2,
                              n_generations=200,
                              crossover_independent_proba=0.5,
                              mutation_independent_proba=0.05,
                              tournament_size=3,
                              n_gen_no_change=10,
                              caching=True,
                              n_jobs=-1)
    selector = selector.fit(X, y)
    y_pred = selector.predict(test_X)
    prediction_data = prediction(test_X, selector)
    ga_acc = cal_accuracy(prediction_data, prediction_data,'GA Algorithm Accuracy,
Classification Report & Confusion Matrix')
    classifier = selector
def runBat():
    text.delete('1.0', END)
    global bat_acc
    train = pd.read_csv(filename)
    alh = BAT(train.values, tf.easom_function, -10, 10, 2, 20)
    data = alh.get_agents()
    X = []
    Y = []
    for i in range(len(data)):
        for j in range(len(data[i])):
            X.append(data[i][j][0:13])
            Y.append(data[i][j][13])

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.1, random_state = 0)
    cls =
RandomForestClassifier(n_estimators=50,max_depth=2,random_state=0,class_weight='b
alanced')
    cls.fit(X_train, y_train)
    prediction_data = prediction(X_test, cls)
    bat_acc = cal_accuracy(y_test, prediction_data,'BAT Algorithm Accuracy,
Classification Report & Confusion Matrix')
def runBee():
    text.delete('1.0', END)
    global bee_acc
    train = pd.read_csv(filename)
    alh = BEE(train.values, tf.easom_function, -10, 10, 2, 20)
    data = alh.get_agents()
    X = []
    Y = []
    for i in range(len(data)):
        for j in range(len(data[i])):
            X.append(data[i][j][0:13])
            Y.append(data[i][j][13])
  X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.1, random_state =
0)
    cls =
RandomForestClassifier(n_estimators=30,max_depth=2,random_state=0,class_weight='b
alanced')
    cls.fit(X_train, y_train)
    prediction_data = prediction(X_test, cls)
    bee_acc = cal_accuracy(y_test, prediction_data,'ABE Algorithm Accuracy,
Classification Report & Confusion Matrix')
def predict():
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="dataset")
    test = pd.read_csv(filename)
    test = test.values[:, 0:12]
    total = len(test)
    text.insert(END,filename+" test file loaded\n");
    y_pred = classifier.predict(test)
    for i in range(len(test)):
        print(str(y_pred[i]))
        if str(y_pred[i]) == '0.0':
            text.insert(END,"X=%s, Predicted = %s" % (test[i], 'No disease detected')+"\n\n")
        if str(y_pred[i]) == '1.0':
           text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Stage 1 Disease
Detected')+"\n\n")
        if str(y_pred[i]) == '2.0':
           text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Stage 2 Disease
Detected')+"\n\n")
        if str(y_pred[i]) == '3.0':
           text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Stage 3 Disease
Detected')+"\n\n")
        if str(y_pred[i]) == '4.0':
           text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Stage 4 Disease
Detected')+"\n\n")
def graph():
    height = [ga_acc,bat_acc,bee_acc]
    bars = ('Genetic Algorithm','Bat Algorithm','Bee Algorithm')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()
def exit():
    main.destroy()
font = ('times', 16, 'bold')
title = Label(main, text='Heart Disease Prediction Using Bio Inspired Algorithms')
title.config(bg='brown', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 14, 'bold')
uploadButton = Button(main, text="Upload Heart Disease", command=upload)
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)
pathlabel = Label(main)
pathlabel.config(bg='brown', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=460,y=100)
geneticButton = Button(main, text="Run Genetic Algorithm",
command=geneticAlgorithm)
geneticButton.place(x=50,y=150)
geneticButton.config(font=font1)
batButton = Button(main, text="Run BAT Algorithm", command=runBat)
batButton.place(x=330,y=150)
batButton.config(font=font1)
beeButton = Button(main, text="Run BEE Algorithm", command=runBee)
beeButton.place(x=620,y=150)
beeButton.config(font=font1)
predictButton = Button(main, text="Upload & Predict Test Data", command=predict)
predictButton.place(x=850,y=150)
predictButton.config(font=font1)
graphButton = Button(main, text="Accuracy Graph", command=graph)
graphButton.place(x=50,y=200)
graphButton.config(font=font1)
exitButton = Button(main, text="Exit", command=exit)
exitButton.place(x=330,y=200)
exitButton.config(font=font1)
font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=250)
text.config(font=font1)
main.config(bg='brown')
main.mainloop()
import random as rn
import numpy as np
from numpy.random import choice as np_choice
class ACO(object):
        def _init_(self, distances, n_ants, n_best, n_iterations, decay, alpha=1, beta=1):
    """
    Args:
        distances (2D numpy.array): Square matrix of distances. Diagonal is assumed to
be np.inf.
        n_ants (int): Number of ants running per iteration
        n_best (int): Number of best ants who deposit pheromone
        n_iteration (int): Number of iterations
        decay (float): Rate it which pheromone decays. The pheromone value is
multiplied by decay, so 0.95 will lead to decay, 0.5 to much faster decay.
        alpha (int or float): exponenet on pheromone, higher alpha gives pheromone more
weight. Default=1
        beta (int or float): exponent on distance, higher beta give distance more weight.
Default=1
    Example:
        ant_colony = AntColony(german_distances, 100, 20, 2000, 0.95, alpha=1, beta=2)
    """
    self.distances  = distances
    self.pheromone = np.ones(self.distances.shape) / len(distances)
    self.all_inds = range(len(distances))
    self.n_ants = n_ants
    self.n_best = n_best
    self.n_iterations = n_iterations
    self.decay = decay
    self.alpha = alpha
    self.beta = beta

  def run(self):
    shortest_path = None
    all_time_shortest_path = ("placeholder", np.inf)
    for i in range(self.n_iterations):
        all_paths = self.gen_all_paths()
        self.spread_pheronome(all_paths, self.n_best, shortest_path=shortest_path)
        shortest_path = min(all_paths, key=lambda x: x[1])
        print (shortest_path)
        if shortest_path[1] < all_time_shortest_path[1]:
            all_time_shortest_path = shortest_path
        self.pheromone * self.decay
    return all_time_shortest_path

  def spread_pheronome(self, all_paths, n_best, shortest_path):
    sorted_paths = sorted(all_paths, key=lambda x: x[1])
    for path, dist in sorted_paths[:n_best]:
        for move in path:
            self.pheromone[move] += 1.0 / self.distances[move]
```

```
 def gen_path_dist(self, path):
        total_dist = 0
        for ele in path:
            total_dist += self.distances[ele]
        return total_dist
def gen_all_paths(self):
        all_paths = []
        for i in range(self.n_ants):
            path = self.gen_path(0)
            all_paths.append((path, self.gen_path_dist(path)))
        return all_paths
def gen_path(self, start):
        path = []
        visited = set()
        visited.add(start)
        prev = start
        for i in range(len(self.distances) - 1):
            move = self.pick_move(self.pheromone[prev], self.distances[prev], visited)
            path.append((prev, move))
            prev = move
            visited.add(move)
        path.append((prev, start)) # going back to where we started
        return path
 def pick_move(self, pheromone, dist, visited):
        pheromone = np.copy(pheromone)
        pheromone[list(visited)] = 0
row = pheromone * self.alpha * (( 1.0 / dist) * self.beta)
norm_row = row / row.sum()
        move = np_choice(self.all_inds, 1, p=norm_row)[0]
        return move
if _name_ == "_main_":
  distances = np.array([[np.inf, 2, 2, 5, 7],
                   [2, np.inf, 4, 8, 2],
                   [2, 4, np.inf, 1, 3],
                   [5, 8, 1, np.inf, 2],
                   [7, 2, 3, 2, np.inf]])
  ant_colony = ACO(distances, 1, 1, 100, 0.95, alpha=1, beta=1)
  shortest_path = ant_colony.run()
  print ("shorted_path: {}".format(shortest_path))
from math import exp
import numpy as np
from random import random
from SwarmPackagePy import intelligence
class BAT(intelligence.sw):
    """
    Bat Algorithm
    """

    def _init_(self, n, function, lb, ub, dimension, iteration, r0=0.9,
              V0=0.5, fmin=0, fmax=0.02, alpha=0.9, csi=0.9):
```

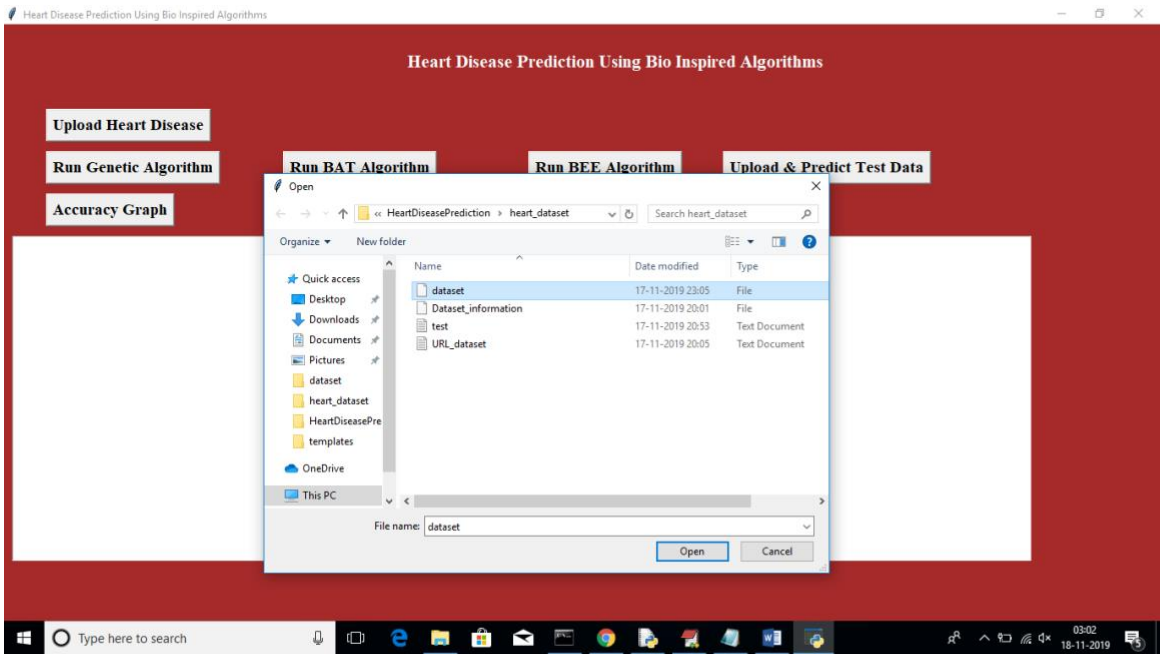
```
        """
        :param n: number of agents
        :param function: test function
        :param lb: lower limits for plot axes
        :param ub: upper limits for plot axes
        :param dimension: space dimension
        :param iteration: number of iterations
        :param r0: level of impulse emission (default value is 0.9)
        :param V0: volume of sound (default value is 0.5)
        :param fmin: min wave frequency (default value is 0)
        :param fmax: max wave frequency (default value is 0.02)
            fmin = 0 and fmax =0.02 - the bests values
        :param alpha: constant for change a volume of sound
         (default value is 0.9)
        :param csi: constant for change a level of impulse emission
         (default value is 0.9)
        """
        super(BAT, self)._init_()
r = [r0 for i in range(len(n))]
self.__agents = n    #np.random.uniform(lb, ub, (n, dimension))
self.points(self._agents
 velocity = np.zeros((len(n), dimension))
        V = [V0 for i in range(len(n))]
         Pbest = self.__agents[np.array([function(i)
                                for i in self.__agents]).argmin()]
        Gbest = Pbest
        f = fmin + (fmin - fmax)
       for t in range(iteration):
            sol = self.__agents
            F = f * np.random.random((len(n), dimension))

        self._set_Gbest(Gbest)

import numpy as np
from random import randint, uniform
import SwarmPackagePy
from SwarmPackagePy import intelligence
class BEE(intelligence.sw):
    """
    Artificial Bee Algorithm
    """
    def _init_(self, n, function, lb, ub, dimension, iteration):
        """
        :param n: number of agents
        :param function: test function
        :param lb: lower limits for plot axes
        :param ub: upper limits for plot axes
        :param dimension: space dimension
        :param iteration: number of iterations
        """
        super(BEE, self)._init_()

        self.__function = function

        self.__agents = n  #np.random.uniform(lb, ub, (n, dimension))
        self.points(self._agents)

        Pbest = self._agents[np.array([function(x) for x in self._agents]).argmin()]
        Gbest = Pbest

        if len(n) <= 10:
            count = n - n // 2, 1, 1, 1
        else:
            a = len(n) // 10
            b = 5
            c = (n - a * b - a) // 2
            d = 2
            count = a, b, c, d

        for t in range(iteration):

            fitness = [function(x) for x in self.__agents]
            sort_fitness = [function(x) for x in self.__agents]
            sort_fitness.sort()
            sort_fitness = np.asarray(sort_fitness)

            best = [self.__agents[i] for i in
                    [fitness.index(x) for x in sort_fitness[:count[0]]]]
            selected = [self.__agents[i]
                        for i in [fitness.index(x)
                                  for x in sort_fitness[1:5]]]

            self._set_Gbest(Gbest)
           def __new(self, l, c, lb, ub):
                        bee = []
           for i in l:
            new = [self.__neighbor(i, lb, ub) for k in range(c)]
            bee += new
        bee += l
         return bee
       def __neighbor(self, who, lb, ub):
       neighbor = np.array(who) + uniform(-1, 1) * (
       np.array(who) - np.array(self._agents[randint(0, len(self._agents) - 1)]))
        neighbor = np.clip(neighbor, lb, ub)

       return list(neighbor)

from _future_ import print_function
import numpy as np
```

```
from sklearn import datasets, linear_model
import pandas as pd
from genetic_selection import GeneticSelectionCV
def main():
    train = pd.read_csv('heart_dataset/dataset')
    test = pd.read_csv('heart_dataset/test.txt')
    test_X = test.values[:, 0:12]
    X = train.values[:, 0:12]
    y = train.values[:, 13]

    estimator = linear_model.LogisticRegression(solver="liblinear", multi_class="ovr")

    selector = GeneticSelectionCV(estimator,
                                  cv=5,
                                  verbose=1,
                                  scoring="accuracy",
                                  max_features=10,
                                  n_population=50,
                                  crossover_proba=0.5,
                                  mutation_proba=0.2,
                                  n_generations=200,
                                  crossover_independent_proba=0.5,
                                  mutation_independent_proba=0.05,
                                  tournament_size=3,
                                  n_gen_no_change=10,
                                  caching=True,
                                  n_jobs=-1)
    selector = selector.fit(X, y)

    print(selector.support_)
    y_pred = selector.predict(test_X)
    print(y_pred)

if _name_ == "_main_":
    main()
```

5.SCREENSHOTS


Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
Run Genetic Algorithm
Run BAT Algorithm
Run BEE Algorithm
Upload & Predict Test Data
Accuracy Graph
Exit
Type here to search
03:01
18-11-2019

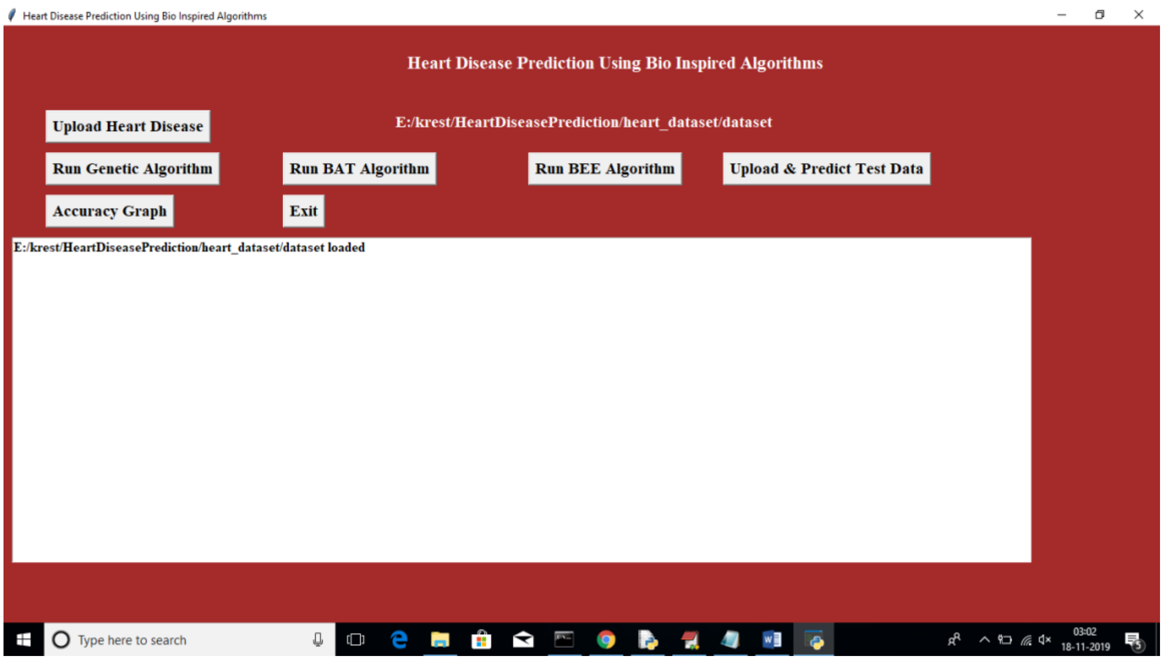
Screenshot 5.1:run.bat file



Name	Date modified	Type
dataset	17-11-2019 23:05	File
Dataset_information	17-11-2019 20:01	File
test	17-11-2019 20:53	Text Document
URL_dataset	17-11-2019 20:05	Text Document

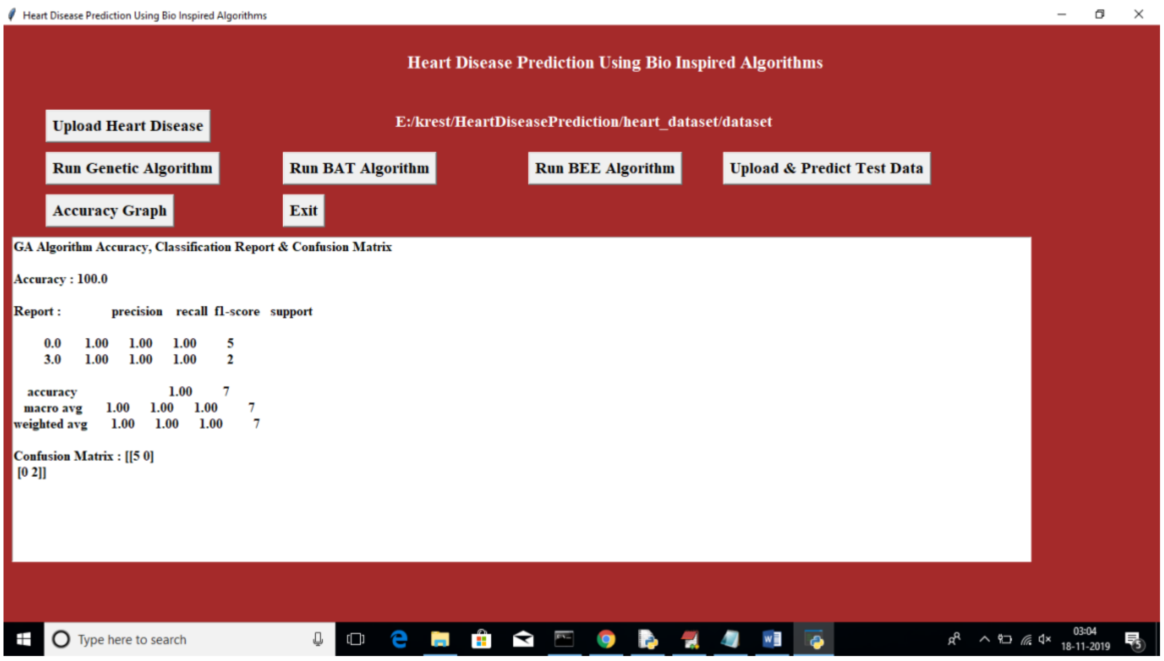
File name: dataset

Screenshot 5.2:Dataset file


Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
E:/krest/HeartDiseasePrediction/heart_dataset/dataset
Run Genetic Algorithm
Run BAT Algorithm
Run BEE Algorithm
Upload & Predict Test Data
Accuracy Graph
Exit

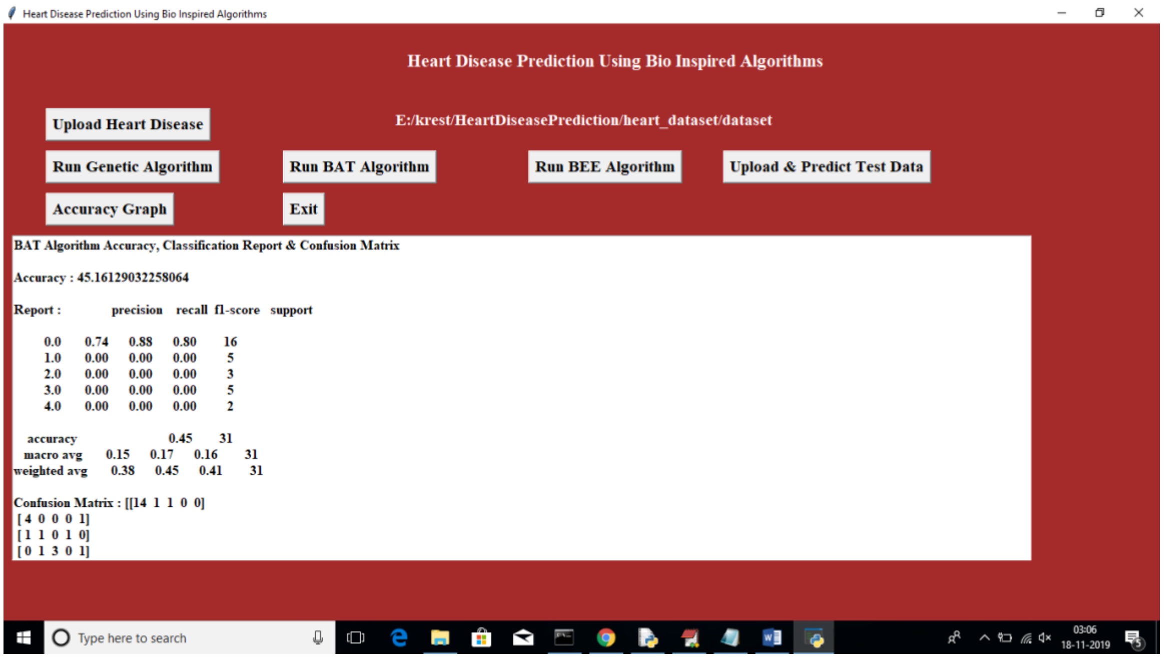
```
E:/krest/HeartDiseasePrediction/heart_dataset/dataset loaded
```

Screenshot 5.3:Dataset file location


Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
E:/krest/HeartDiseasePrediction/heart_dataset/dataset
Run Genetic Algorithm
Run BAT Algorithm
Run BEE Algorithm
Upload & Predict Test Data
Accuracy Graph
Exit

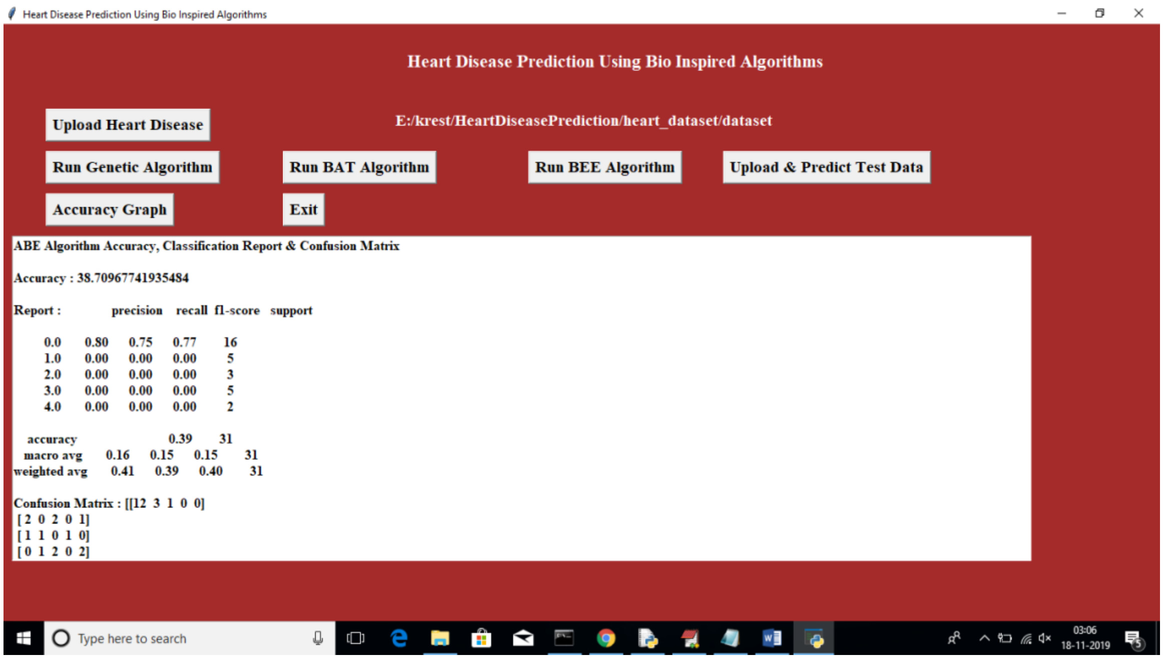
```
GA Algorithm Accuracy, Classification Report & Confusion Matrix

Accuracy : 100.0

Report :            precision    recall  f1-score   support

        0.0       1.00      1.00      1.00         5
        3.0       1.00      1.00      1.00         2

    accuracy                           1.00         7
   macro avg       1.00      1.00      1.00         7
weighted avg       1.00      1.00      1.00         7

Confusion Matrix : [[5 0]
 [0 2]]
```

Screenshot 5.4: Run Genetic Algorithm


Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
E:/krest/HeartDiseasePrediction/heart_dataset/dataset
Run Genetic Algorithm
Run BAT Algorithm
Run BEE Algorithm
Upload & Predict Test Data
Accuracy Graph
Exit
Type here to search
03:06
18-11-2019

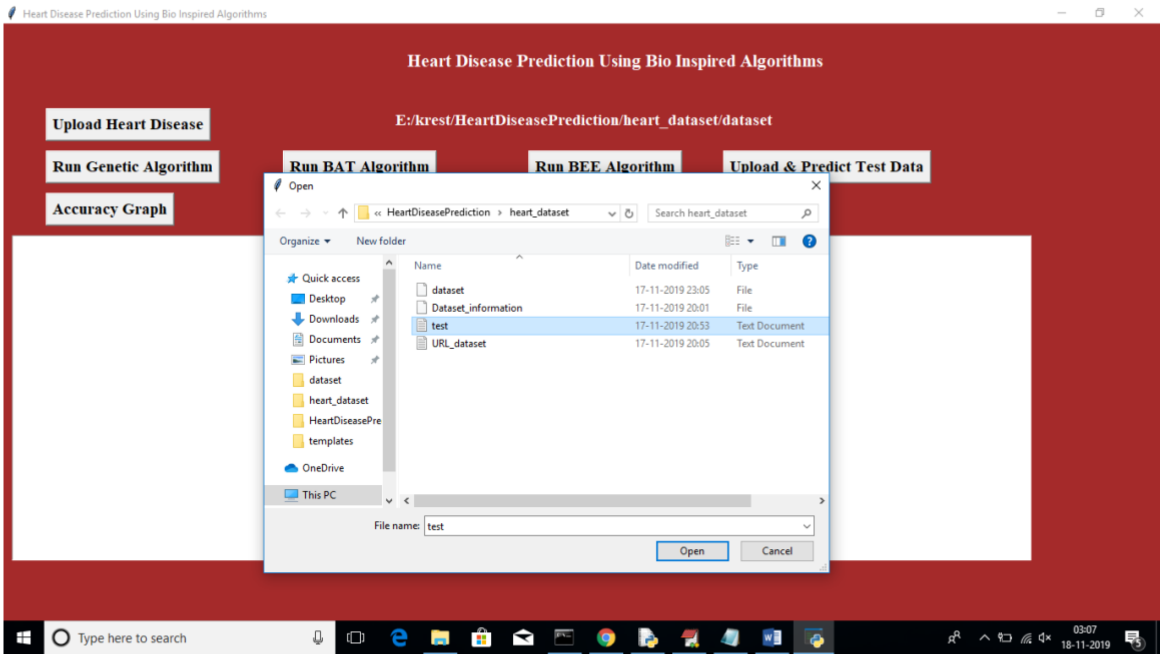
```
BAT Algorithm Accuracy, Classification Report & Confusion Matrix

Accuracy : 45.16129032258064

Report :           precision    recall  f1-score   support

      0.0       0.74      0.88      0.80        16
      1.0       0.00      0.00      0.00         5
      2.0       0.00      0.00      0.00         3
      3.0       0.00      0.00      0.00         5
      4.0       0.00      0.00      0.00         2

   accuracy                           0.45        31
  macro avg       0.15      0.17      0.16        31
weighted avg       0.38      0.45      0.41        31

Confusion Matrix : [[14  1  1  0  0]
 [ 4  0  0  0  1]
 [ 1  1  0  1  0]
 [ 0  1  3  0  1]
```

Screenshot 5.5:Run BAT Algorithm

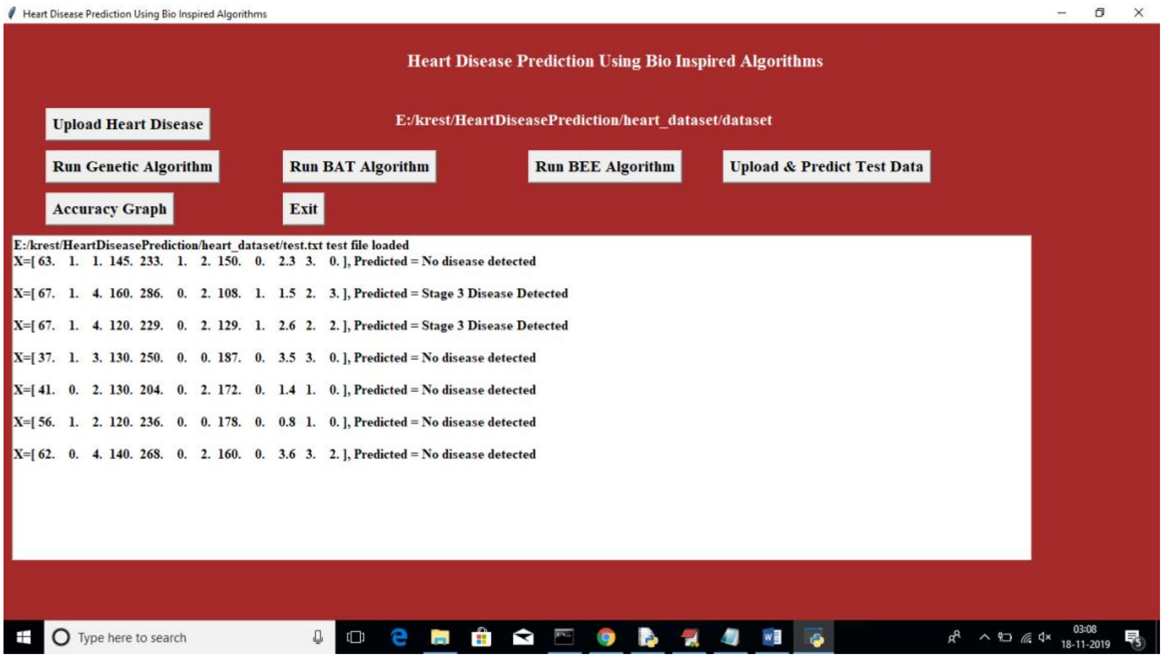

Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
E:/krest/HeartDiseasePrediction/heart_dataset/dataset
Run Genetic Algorithm
Run BAT Algorithm
Run BEE Algorithm
Upload & Predict Test Data
Accuracy Graph
Exit
Type here to search
03:06
18-11-2019

```
ABE Algorithm Accuracy, Classification Report & Confusion Matrix

Accuracy : 38.70967741935484

Report :           precision    recall  f1-score   support

      0.0       0.80      0.75      0.77        16
      1.0       0.00      0.00      0.00         5
      2.0       0.00      0.00      0.00         3
      3.0       0.00      0.00      0.00         5
      4.0       0.00      0.00      0.00         2

   accuracy                           0.39        31
  macro avg       0.16      0.15      0.15        31
weighted avg       0.41      0.39      0.40        31

Confusion Matrix : [[12  3  1  0  0]
 [ 2  0  2  0  1]
 [ 1  1  0  1  0]
 [ 0  1  2  0  2]
```

Screenshot 5.6:Run BEE Algorithm

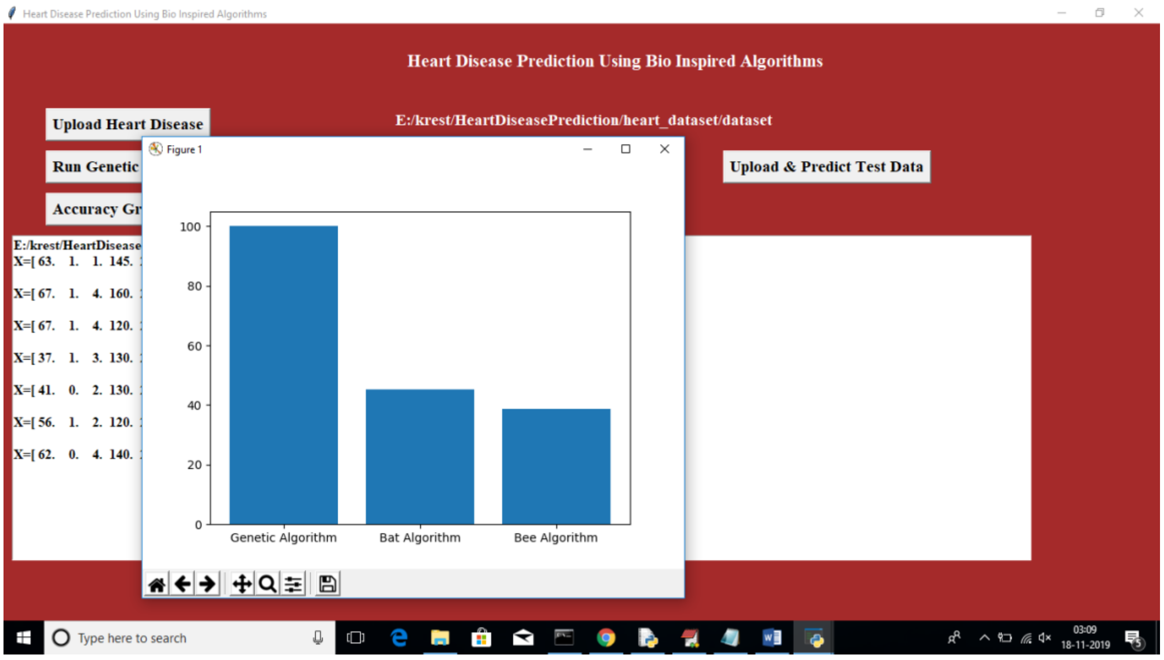

Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
E:/krest/HeartDiseasePrediction/heart_dataset/dataset
Run Genetic Algorithm
Run BAT Algorithm
Run BEE Algorithm
Upload & Predict Test Data
Accuracy Graph
Open
HeartDiseasePrediction › heart_dataset
Search heart_dataset
Organize
New folder
Quick access
Desktop
Downloads
Documents
Pictures
dataset
heart_dataset
HeartDiseasePre
templates
OneDrive
This PC
Name
Date modified
Type
dataset 17-11-2019 23:05 File
Dataset_information 17-11-2019 20:01 File
test 17-11-2019 20:53 Text Document
URL_dataset 17-11-2019 20:05 Text Document
File name: test
Open
Cancel
Type here to search
03:07
18-11-2019

Screenshot 5.7:Test file


Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
E:/krest/HeartDiseasePrediction/heart_dataset/dataset
Run Genetic Algorithm
Run BAT Algorithm
Run BEE Algorithm
Upload & Predict Test Data
Accuracy Graph
Exit
Type here to search
03:08
18-11-2019

```
E:/krest/HeartDiseasePrediction/heart_dataset/test.txt test file loaded
X=[ 63.   1.   1. 145. 233.   1.   2. 150.   0.   2.3  3.   0. ], Predicted = No disease detected

X=[ 67.   1.   4. 160. 286.   0.   2. 108.   1.   1.5  2.   3. ], Predicted = Stage 3 Disease Detected

X=[ 67.   1.   4. 120. 229.   0.   2. 129.   1.   2.6  2.   2. ], Predicted = Stage 3 Disease Detected

X=[ 37.   1.   3. 130. 250.   0.   0. 187.   0.   3.5  3.   0. ], Predicted = No disease detected

X=[ 41.   0.   2. 130. 204.   0.   2. 172.   0.   1.4  1.   0. ], Predicted = No disease detected

X=[ 56.   1.   2. 120. 236.   0.   0. 178.   0.   0.8  1.   0. ], Predicted = No disease detected

X=[ 62.   0.   4. 140. 268.   0.   2. 160.   0.   3.6  3.   2. ], Predicted = No disease detected
```

Screenshot 5.8:Heart Disease Prediction


Heart Disease Prediction Using Bio Inspired Algorithms
Heart Disease Prediction Using Bio Inspired Algorithms
Upload Heart Disease
E:/krest/HeartDiseasePrediction/heart_dataset/dataset
Run Genetic
Accuracy Gr
Upload & Predict Test Data
Figure 1
100
80
60
40
20
0
Genetic Algorithm
Bat Algorithm
Bee Algorithm
Type here to search
03:09
18-11-2019

```
E:/krest/HeartDisease
X=[ 63.   1.   1.  145.
X=[ 67.   1.   4.  160.
X=[ 67.   1.   4.  120.
X=[ 37.   1.   3.  130.
X=[ 41.   0.   2.  130.
X=[ 56.   1.   2.  120.
X=[ 62.   0.   4.  140.
```

Screenshot 5.9:Accurracy Graph

6.TESTING

6.1 INTRODUCTION TO TESTING

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, 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.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. 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.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully 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.

6.2.3 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.

7.CONCLUSION&FUTURE SCOPE

7.1 PROJECT CONCLUSION

In conclusion, our methodology for heart disease prediction using bio-inspired algorithms yielded promising results. Among the bio-inspired algorithms tested, the Genetic Algorithm (GA) demonstrated higher accuracy compared to the Bat Algorithm (BA) and Bee Algorithm (BA). The integration of bio-inspired algorithms into the prediction model improved its performance, surpassing models without these algorithms. Further research and exploration of different bio-inspired algorithms and hybrid approaches are recommended for enhancing accuracy. By continuously improving prediction models with bio-inspired algorithms, we can make significant advancements in early detection and improve patient care in heart disease.

7.2 FUTURE SCOPE

Looking ahead, there are several exciting opportunities for expanding and refining the methodology for heart disease prediction using bio-inspired algorithms. Some potential areas of future research include:

- Exploration of alternative bio-inspired algorithms and their integration with the heart disease prediction model.
- Investigation of hybrid approaches combining multiple bio-inspired algorithms or integrating them with other machine learning techniques.
- Incorporation of deep learning models to enhance prediction accuracy andinterpretability.
- Integration of big data and real-time monitoring for more robust and accurate predictions.
- Evaluation of the clinical utility and real-world applicability of the developed models.
- Focus on interpretability and explainability to gain trust and acceptance in the medical community.
- Generalization of the methodology to other diseases for broader healthcare applications.

8. REFERENCES

8.1 REFERENCES

[1] "Disease Prediction by Machine Learning Over Big Data from Healthcare Communities" by Min Chen. Published in: IEEE Access (Volume: 5) https://ieeexplore.ieee.org/document/7912315

[2] Interactive Atlas of Heart Disease and Stroke Article for Heart Disease Statistics https://www.cdc.gov/heartdisease/facts.htm

[3]Heart disease dataset from Kaggle https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset This directory contains 4 databases concerning heart disease diagnosis. All attributes are numeric-valued.

[4] K.Prasanna Lakshmi, Dr. C.R.K.Reddy, "Fast Rule-Based Heart Disease Prediction using Associative Classification Mining", IEEE International Conference on Computer, Communication and Control (IC4-2015).

[5] B. Qian, X. Wang, N. Cao, H. Li, and Y.-G. Jiang, ''A relative similarity based method for interactive patient risk prediction,'' Data Mining Knowl. Discovery, vol. 29, no. 4, pp. 1070–1093, 2015.

[6] Senthilkumar Mohan, Chandrasegar Thirumalai, and Gautam Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques", IEEE Access 2019.

8.2 GITHUBLINK

https://github.com/ashwitha10/detecting-mental-disorders