**Disease Prediction by Machine Learning Over Big Data From Healthcare Communities**

**1. INTRODUCTION**

According to a report by McKinsey, 50% of Americans have one or more chronic diseases, and 80% of American medical care fee is spent on chronic disease treatment. With the improvement of living standards, the incidence of chronic disease is increasing. The United States has spent an average of 2.7 trillion USD annually on chronic disease treatment. This amount comprises 18% of the entire annual GDP of the United States. The healthcare problem of chronic diseases is also very important in many other countries. In China, chronic diseases are the main cause of death, according to a Chinese report on nutrition and chronic diseases in 2015, 86.6% of deaths are caused by chronic diseases. Therefore, it is essential to perform risk assessments for chronic dis- eases. With the growth in medical data, collecting electronic health records (EHR) is increasingly convenient. Besides, rst presented a bio-inspired high-performance heterogeneous vehicular telematics paradigm, such that the collection of mobile users' health-related real-time big data can be achieved with the deployment of advanced heterogeneous vehicular networks. Chen et al. proposed a healthcare system using smart clothing for sustainable health monitoring. Qiu et al. had thoroughly studied the heterogeneous systems and achieved the best results for cost minimization on tree and simple path cases for heterogeneous systems. Patients' statistical information, test results and disease history are recorded in the EHR, enabling us to identify potential data-centric solutions to reduce the costs of medical case studies. Wang et al. proposed an efficient estimating algorithm for the telehealth cloud system and designed a data coherence protocol for the PHR(Personal Health Record)-based distributed system. Bates et al. proposed six applications of big data in the field of health- care. Qiu et al. proposed an optimal big data sharing algorithm to handle the complicate data set in telehealth with cloud techniques. One of the applications is to identify high-risk patients which can be utilized to reduce medical cost since high-risk patients often require expensive healthcare. Moreover, in the rst paper proposing health- care cyber-physical system, it innovatively brought for- ward the concept of prediction-based healthcare applications, including health risk assessment. Prediction using traditional disease risk models usually involves a machine learning algorithm (e.g., logistic regression and regression analysis, etc.), and especially a supervised learning algorithm by the use of training data with labels to train the model. In the test set, patients can be classied into groups of either high-risk or low-risk. These models are valuable in clinical situations and are widely studied. However, these schemes have the following characteristics and defects. The data set is typically small, for patients and diseases with specic conditions, the characteristics are selected through experience. However, these pre-selected characteristics maybe not satisfy the changes in the disease and its influencing factors. With the development of big data analytics technology, more attention has been paid to disease prediction from the perspective of big data analysis, various researches have been conducted by selecting the characteristics automatically from a large number of data to improve the accuracy of risk classification, rather than the previously selected characteristics. However, those existing work mostly considered structured data. For unstructured data, for example, using convolutional neural network (CNN) to extract text characteristics automatically has already attracted wide attention and also achieved very good results. However, to the best of our knowledge, none of previous work handle Chinese medical text data by CNN. Furthermore, there is a large difference between diseases in different regions, primarily because of the diverse climate and living habits in the region. Thus, risk classification based on big data analysis, the following challenges remain: How should the missing data be addressed? How should the main chronic diseases in a certain region and the main characteristics of the disease in the region be determined? How can big data analysis technology be used to analyze the disease and create a better model? To solve these problems, we combine the structured and unstructured data in healthcare field to assess the risk of disease. First, we used latent factor model to reconstruct the missing data from the medical records collected from a hospital in central China. Second, by using statistical knowledge, we could determine the major chronic diseases in the region. Third, to handle structured data, we consult with hospital experts to extract useful features. For unstructured text data, we select the features automatically using CNN algorithm. Finally, we propose a novel CNN-based multimodal disease risk prediction (CNN-MDRP) algorithm for structured and unstructured data. The disease risk model is obtained by the combination of structured and unstructured features. Through the experiment, we draw a conclusion that the performance of CNN-MDPR is better than other existing methods.

**1.1 Objective of the Project**

The healthcare problem of chronic diseases is also very important in many other countries. Therefore, it is essential to perform risk assessments for chronic diseases. In this paper, we propose a new convolutional neural network based multimodal disease risk prediction (CNN-MDRP) algorithm using structured and unstructured data from hospital. Compared to several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed which is faster than that of the CNN-based unimodal disease risk prediction (CNN-UDRP) algorithm.

**2. LITERATURE SURVEY**

**``A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics,''**

With the increasing demands for vehicle-to-vehicle and vehicle-to-infrastructure communications in intelligent transportation systems, new generation of vehicular telematics inevitably depends on the cooperation of heterogeneous wireless networks. In heterogeneous vehicular telematics, the network selection is an important step to the realization of multimode communications that use multiple access technologies and multiple radios in a collaborative manner. This paper presents an innovative network selection solution for the fundamental technological requirement of multimode communications in heterogeneous vehicular telematics. To guarantee the QoS satisfaction of multiple mobile users and the efficient utilization and fair allocation of heterogeneous network resources in a global sense, a dynamic and self-adaptive method for network selection is proposed. It is biologically inspired by the cellular gene network, which enables terminals to dynamically select an appropriate access network according to the variety of QoS requirements and to the dynamic conditions of various available networks. The experimental results prove the effectiveness of the bioinspired scheme and confirm that the proposed network selection method provides better global performance when compared with the utility function method with greedy optimization.

**`Wearable 2.0: Enable human-cloud integration in next generation healthcare system,''**

With the rapid development of the Internet of Things, cloud computing, and big data, more comprehensive and powerful applications become available. Meanwhile, people pay more attention to higher QoE and QoS in a “terminal- cloud” integrated system. Specifically, both advanced terminal technologies (e.g., smart clothing) and advanced cloud technologies (e.g., big data analytics and cognitive computing in clouds) are expected to provide people with more reliable and intelligent services. Therefore, in this article we propose a Wearable 2.0 healthcare system to improve QoE and QoS of the next generation healthcare system. In the proposed system, washable smart clothing, which consists of sensors, electrodes, and wires, is the critical component to collect users' physiological data and receive the analysis results of users' health and emotional status provided by cloud-based machine intelligence.

**``Emotion communication system,''**

In today's increasingly rich material life, people are shifting their focus from the physical world to the spiritual world. In order to identify and care for people's emotions, human-machine interaction systems have been created. The currently available human-machine interaction systems often support the interaction between human and robot under the line-of-sight (LOS) propagation environment, while most communications in terms of human-to-human and human-to-machine are non-LOS (NLOS). In order to break the limitation of the traditional human-machine interaction system, we propose the emotion communication system based on NLOS mode. Specifically, we first define the emotion as a kind of multimedia which is similar to voice and video. The information of emotion can not only be recognized, but can also be transmitted over a long distance. Then, considering the real-time requirement of the communications between the involved parties, we propose an emotion communication protocol, which provides a reliable support for the realization of emotion communications. We design a pillow robot speech emotion communication system, where the pillow robot acts as a medium for user emotion mapping. Finally, we analyze the real-time performance of the whole communication process in the scene of a long distance communication between a mother-child users' pair, to evaluate the feasibility and effectiveness of emotion communications.

**``Optimal big data sharing approach for tele- health in cloud computing,'**

The rapid development of tele-health systems have received driving engagements from various emerging techniques, such as big data and cloud computing. Sharing data among multiple tele-health systems is an adaptive approach for improving service quality via the network-based technologies. However, current implementations of data sharing in cloud computing is still facing the restrictions caused by the networking capacities and virtual machine switches. In this paper, we focus on the problem of data sharing obstacles in cloud computing and propose an approach that uses dynamic programming to produce optimal solutions to data sharing mechanisms. The proposed approach is called Optimal Telehealth Data Sharing Model (OTDSM), which considers transmission probabilities, maximizing network capacities, and timing constraints. Our experimental results have proved the flexibility and adoptability of the proposed method.

**``Localization based on social big data analysis in the vehicular networks,'**

Location-based services, especially for vehicular localization, are an indispensable component of most technologies and applications related to the vehicular networks. However, because of the randomness of the vehicle movement and the complexity of a driving environment, attempts to develop an effective localization solution face certain difficulties. In this paper, an overlapping and hierarchical social clustering model (OHSC) is first designed to classify the vehicles into different social clusters by exploring the social relationship between them. By using the results of the OHSC model, we propose a social-based localization algorithm (SBL) that use location prediction to assist in global localization in the vehicular networks. The experiment results validate the performance of the OHSC model and show that the presented SBL algorithm demonstrates superior localization performance compared with the existing methods.

**``A manufacturing big data solution for active preventive maintenance,''**

Industry 4.0 has become more popular due to recent developments in cyber-physical systems, big data, cloud computing, and industrial wireless networks. Intelligent manufacturing has produced a revolutionary change, and evolving applications, such as product lifecycle management, are becoming a reality. In this paper, we propose and implement a manufacturing big data solution for active preventive maintenance in manufacturing environments. First, we provide the system architecture that is used for active preventive maintenance. Then, we analyze the method used for collection of manufacturing big data according to the data characteristics. Subsequently, we perform data processing in the cloud, including the cloud layer architecture, the real-time active maintenance mechanism, and the offline prediction and analysis method. Finally, we analyze a prototype platform and implement experiments to compare the traditionally used method with the proposed active preventive maintenance method. The manufacturing big data method used for active preventive maintenance has the potential to accelerate implementation of Industry 4.0.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

In existing systems, the analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease outbreaks.

**Disadvantages of Existing System:**

1. Accuracy is less.

**3.2 Proposed System**

In this paper, We propose a new convolutional neural network (CNN)-based multimodal disease risk prediction algorithm using structured and unstructured data from hospital. To the best of our knowledge, none of the existing work focused on both data types in the area of medical big data analytics. Compared with several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed, which is faster than that of the CNN-based unimodal disease risk prediction algorithm.

**Advantages:**

1. Accuracy is more.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loa ded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms *what* must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .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:
* **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.

* **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. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

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.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**Operating Environment**

Windows XP.

**HARDWARE REQUIREMENTS:**

# Processor - Pentium –IV

* Speed - 1.1 Ghz
* RAM - 256 MB(min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows7/8
* Programming Language - Python

**4. SYSTEM DESIGN**

**UML Diagram:**

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

**Class Diagram:**



**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

**Use case Diagram:**

Upload Heart disease

Run Genetic Algorithm

Run BAT Algorithm

Run BEE Algorithm

Upload & predict test data

Exit

User

Accuracy Graph

**Sequence diagram:**

A **sequence diagram** 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. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes 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. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.



**Component Diagram:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

**Component Diagram :**

User

Upload Heart Disease

Run Genetic Algorithm

Run BAT

Algorithm

Run BEE

Algorithm

Upload & predict

Test data

Accuracy Graph

**Deployment Diagram:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

**Deployment Diagram:**

User

Upload Heart

Disease

Run Genetic

Algorithm

Run BAT

Algorithm

Accuracy Graph

Run BEE

Algorithm

Upload & predict

Test data

**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

Activity Diagram:

Upload heart disease

Run Genetic Algorithm

Run BAT Algorithm

Run BEE Algorithm

Upload & predict test data

Accuracy Graph

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

2. Heart disease loaded

Patient

4. Accuracy displayed

1. Upload heart disease 6. Accuracy displayed

3. Run Genetic Algorithm 8. Accuracy displayed

5. Run BAT Algorithm 10. Test data loaded

7. Run BEE Algorithm 12. Accuracy graph displayed

9. Upload & predict test data

11. Accuracy Graph

**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

**Main.py:**

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

# Function to calculate accuracy

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='balanced')

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='balanced')

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()

**GA.py:**

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()

**ACO.py:**

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))

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

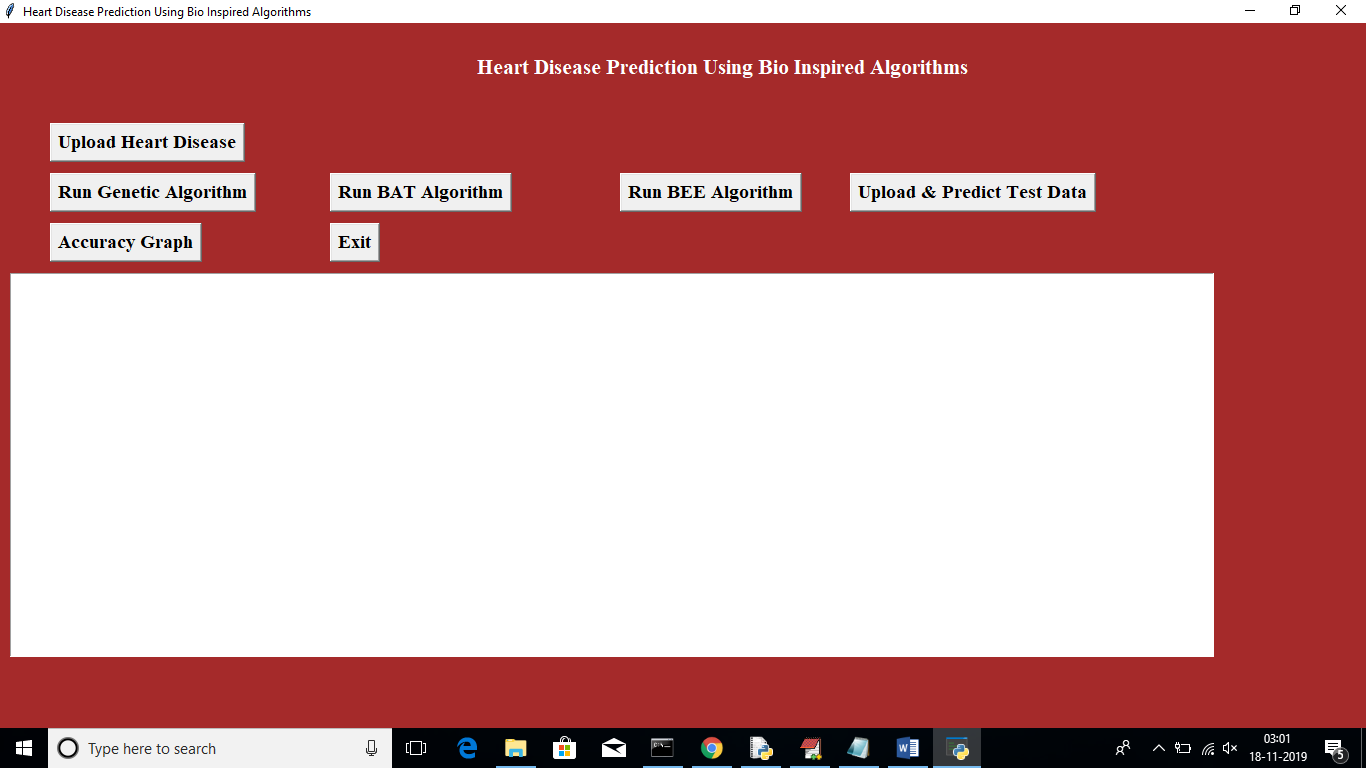
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

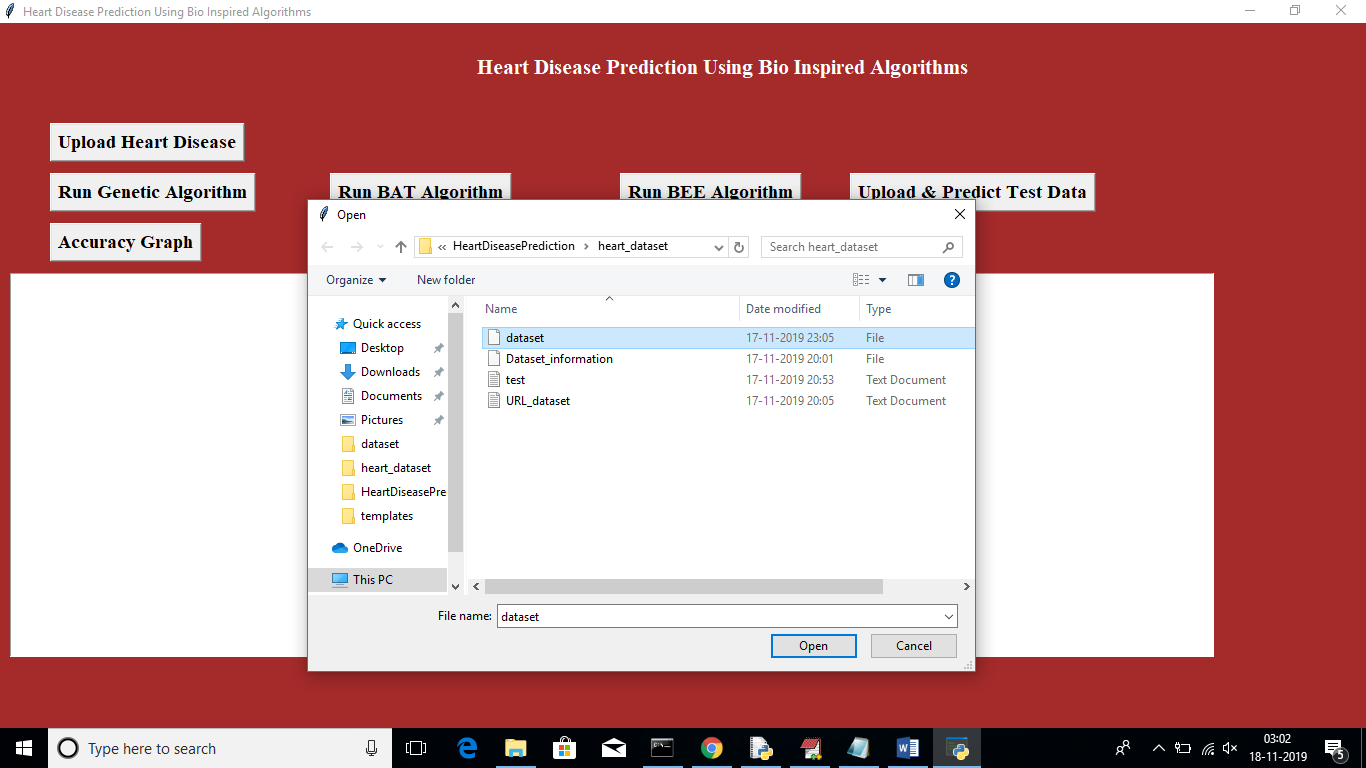
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload dataset | Test whether the dataset is uploaded or not | if the dataset may not uploaded | we cannot do any further operations | we can do further operations | High | High |
| 02 | genetic algorithm | Verify the genetic algorithm will run or not | Without loading the dataset | we cannot run genetic algorithm | accuracy is displayed | High | High |
| 03 | BAT Algorithm | Verify the BAT algorithm will run or not | Without loading the dataset | we cannot run BAT algorithm | accuracy is displayed | High | High |
| 04 | BEE Algorithm | Verify the BEE algorithm will run or not | Without loading the dataset | we cannot run BEE algorithm | accuracy is displayed | High | High |
| 05 | Upload & predict test data | Verify either the details are displaying or not | without loading the predict test data | we cannot predict the test data | the test data predicted successfully | High | High |
| 06 | Accuracy Graph | verify the accuracy graph is displayed or not | without saving the accuracy values of each algorithms | we cannot get accuracy graph | accuracy graph is displayed | High | High |

**7. SCREENSHOTS**

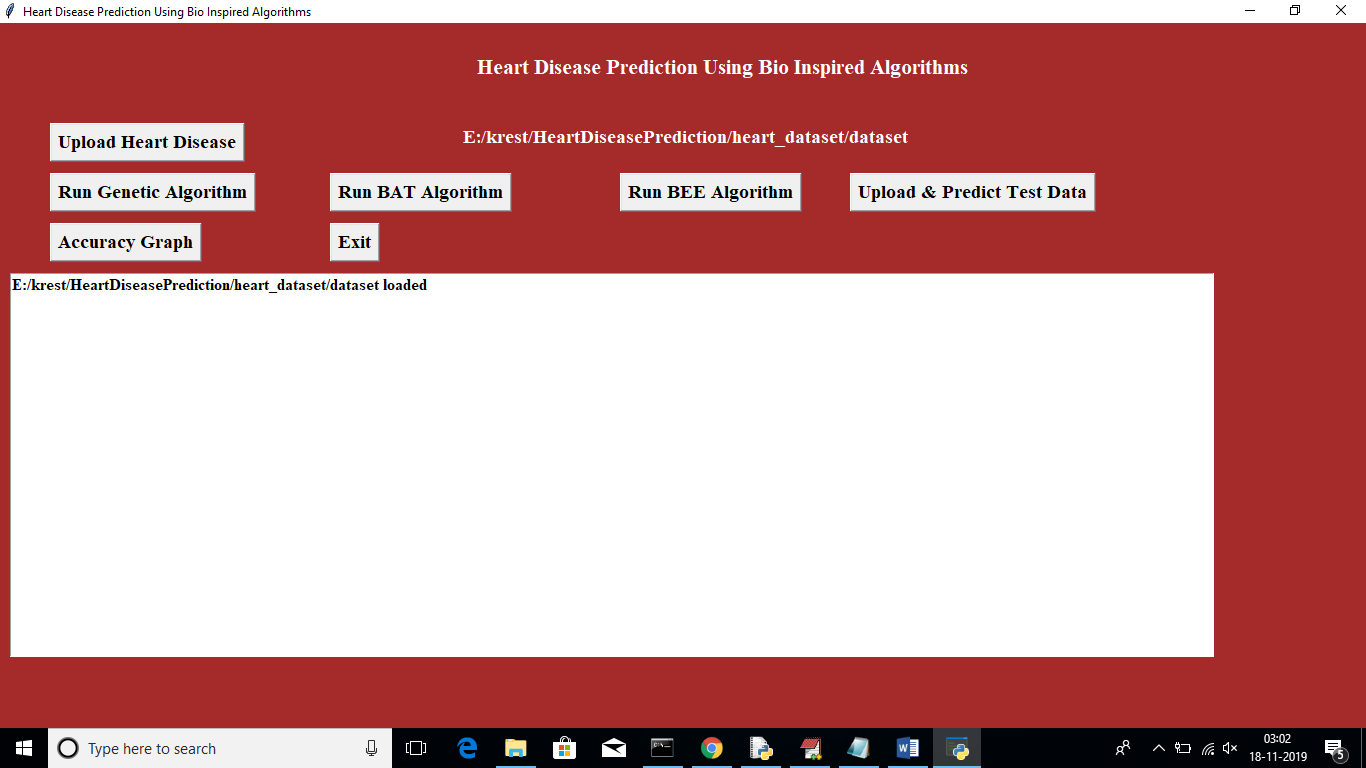
To run this project double click on ‘run.bat’ file to get below screen



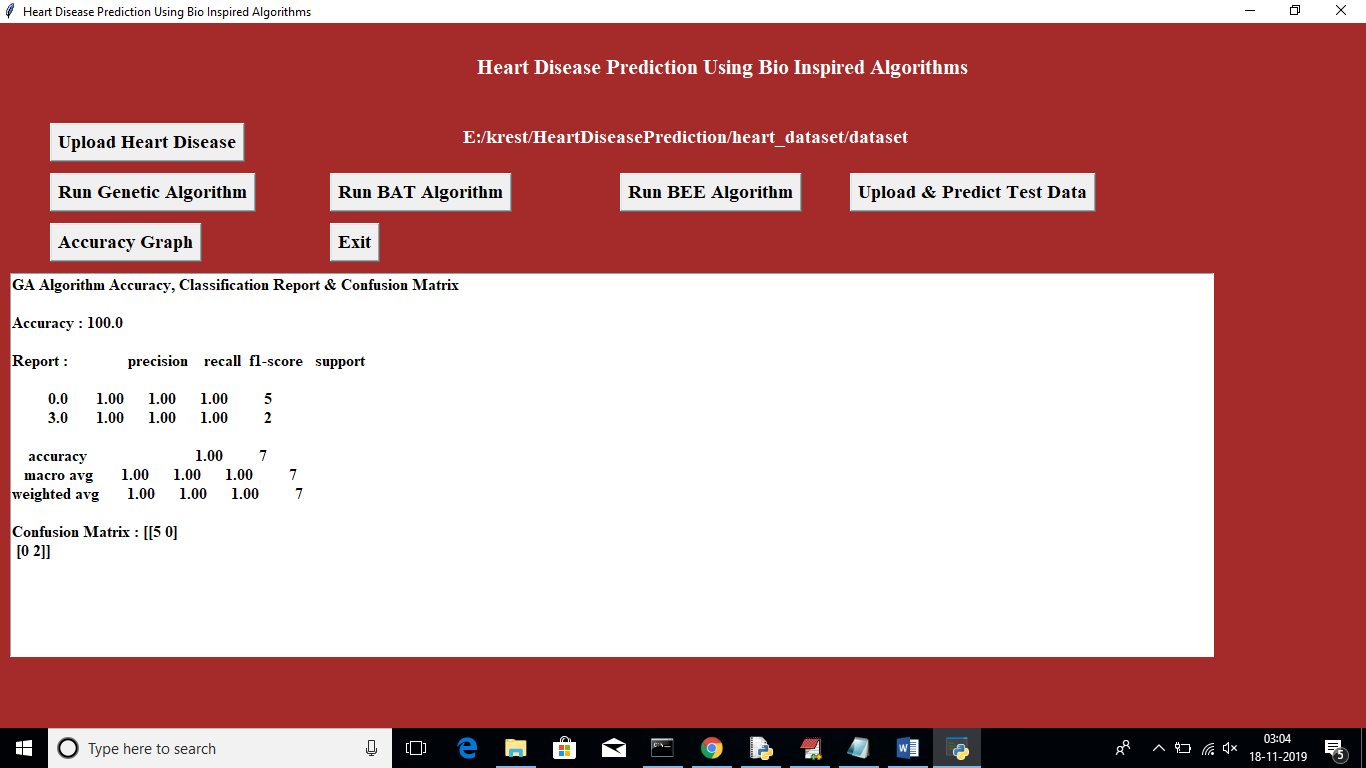
In above screen click on ‘Upload Heart Disease’ button and upload heart disease dataset. See below screen



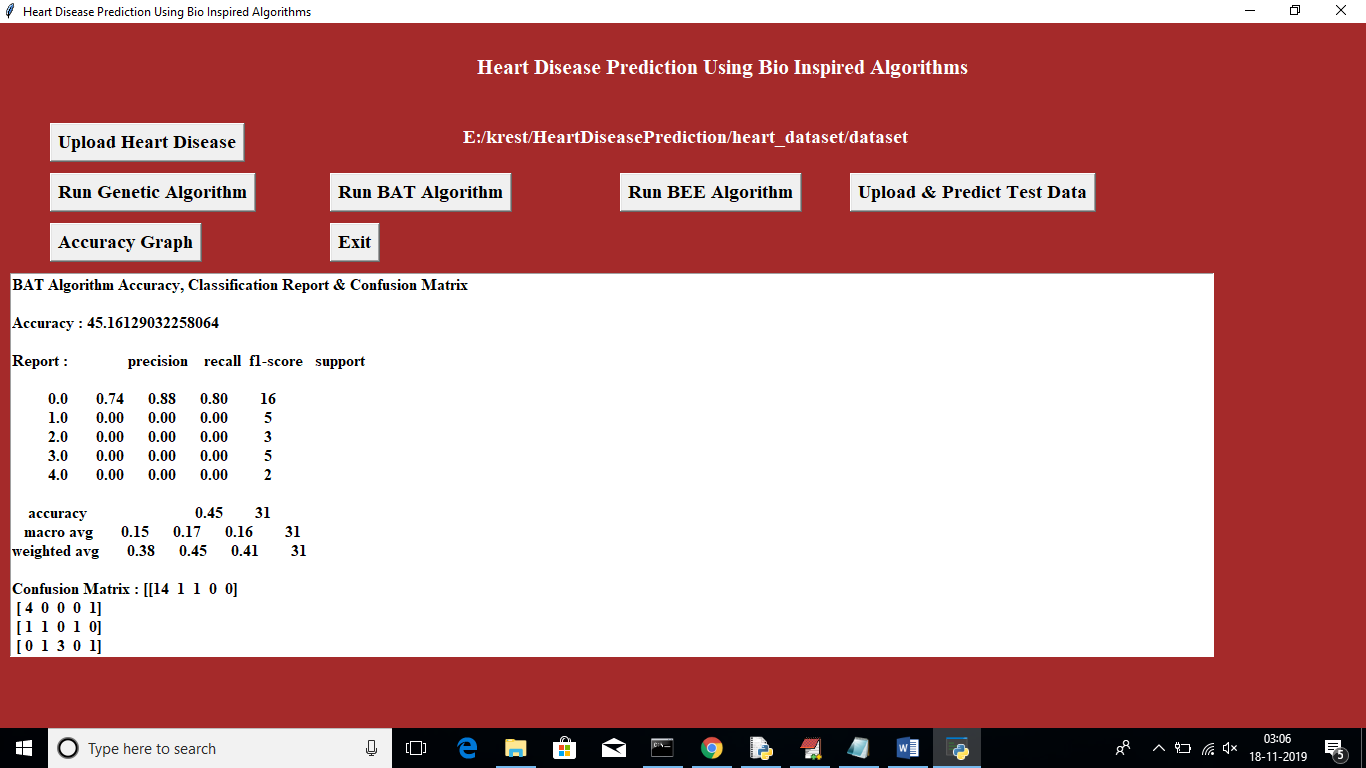
In above screen uploading dataset file, after uploading will get below screen



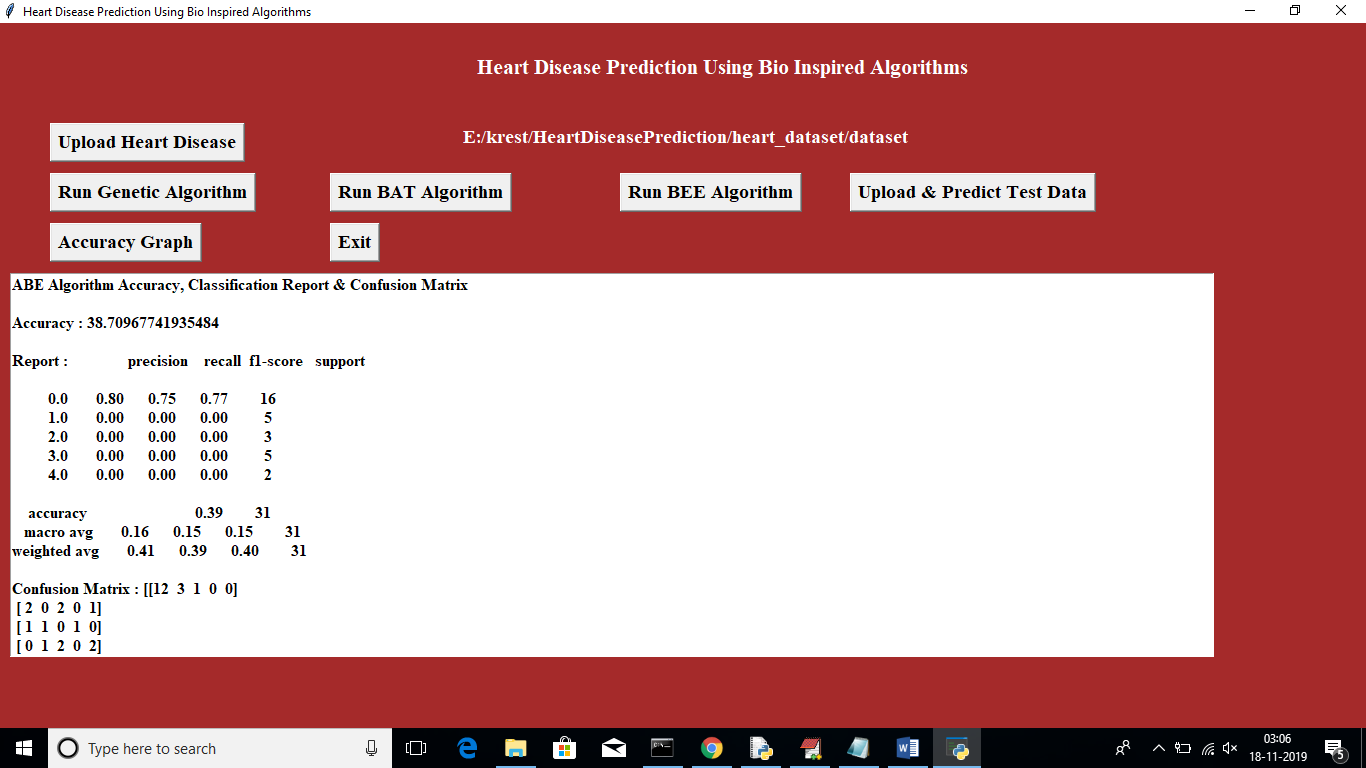
Now click on ‘Run Genetic Algorithm’ button to run genetic algorithm on dataset and to get its accuracy details. While running this algorithm u can see black console to see feature selection process, while running it will open empty windows, u just close all those empty windows except current window



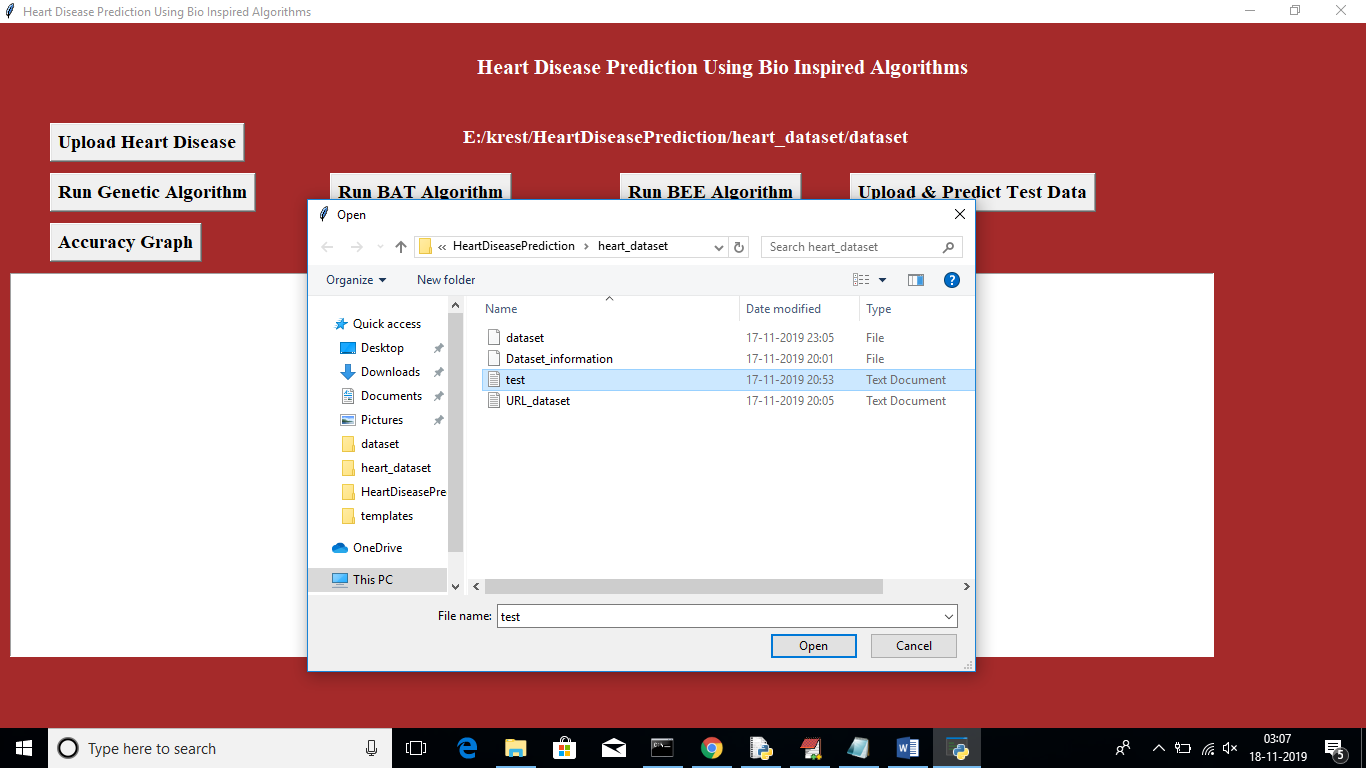
In above screen for GA accuracy, precision and recall we got 100% result. Now click on ‘Run Bat’ algorithm button to get its accuracy



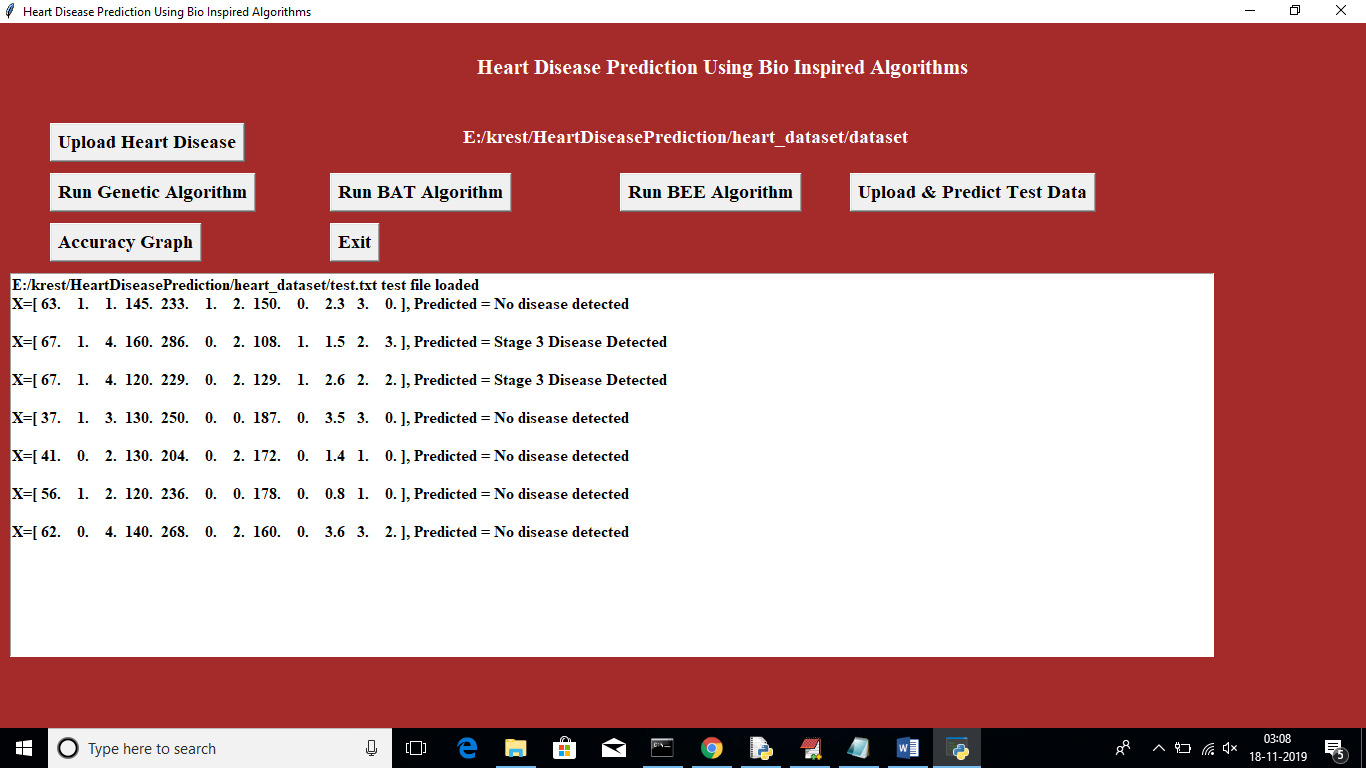
In above screen for BAT we got 45% accuracy, now click on ‘Run BEE Algorithm’ button to get BEE accuracy



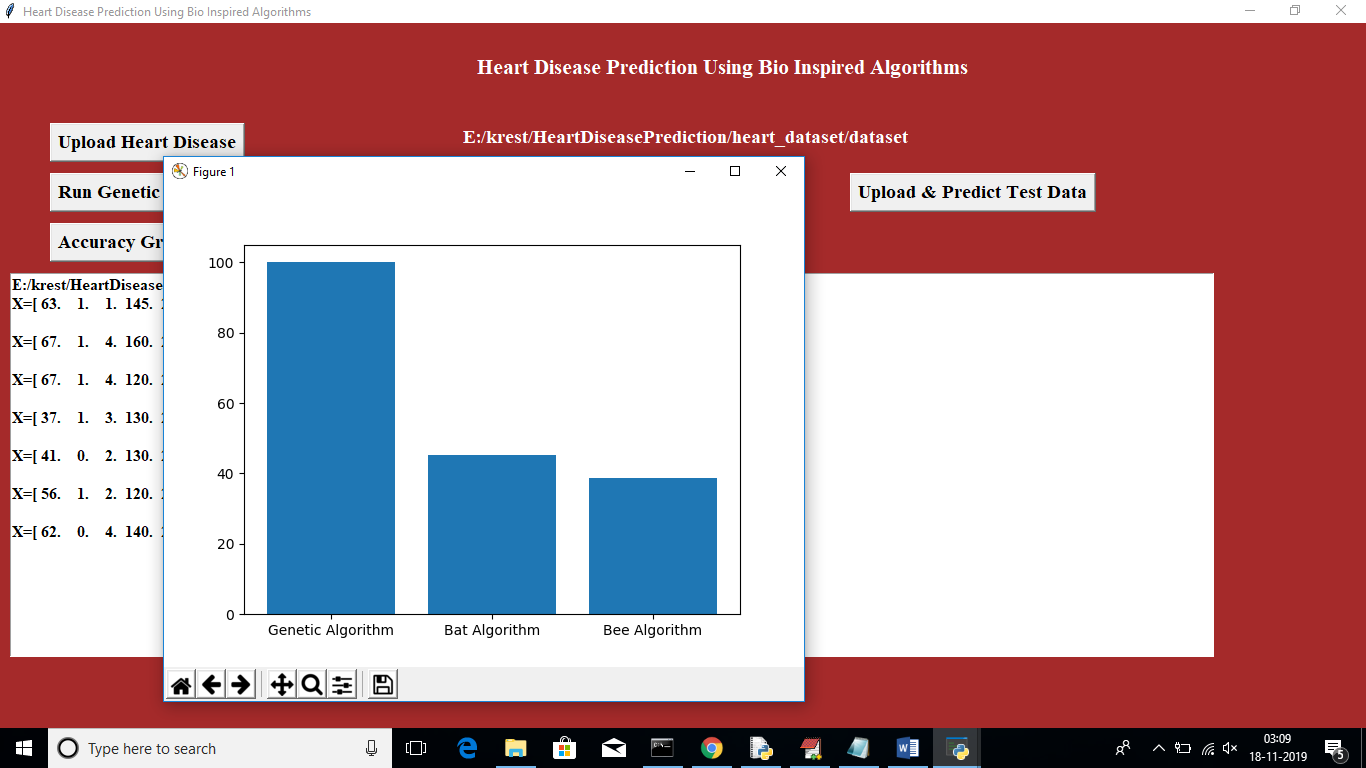
In above screen for BEE we got 38% accuracy, now click on ‘Upload & Predict Test Data’ button to upload test data and to predict it class



In above screen I am uploading test file which contains test data without class label, after uploading test data will get below screen



In above screen application has predicted disease stages. Now click on ‘Accuracy Graph’ button to view accuracy of all algorithms in graph format



In above graph x-axis represents Algorithm Name and y-axis represents accuracy of those algorithms

**8. CONCLUSION**

In this paper, we propose a new convolutional neural network based multimodal disease risk prediction (CNN-MDRP) algorithm using structured and unstructured data from hos- pital. To the best of our knowledge, none of the existing work focused on both data types in the area of medical big data analytics. Compared to several typical prediction algo- rithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed which is faster than that of the CNN-based unimodal disease risk prediction (CNN-UDRP) algorithm.

**9. REFERENCES**

[1] P. Groves, B. Kayyali, D. Knott, and S. van Kuiken, The`Big Data'Revolution in Healthcare: Accelerating Value and Innovation. USA: Center for US Health System Reform Business Technology Ofce, 2016.

[2] M. Chen, S. Mao, and Y. Liu, ``Big data: A survey,'' Mobile Netw. Appl., vol. 19, no. 2, pp. 171209, Apr. 2014.

[3] P. B. Jensen, L. J. Jensen, and S. Brunak, ``Mining electronic health records: Towards better research applications and clinical care,'' Nature Rev. Genet., vol. 13, no. 6, pp. 395405, 2012.

[4] D. Tian, J. Zhou, Y. Wang, Y. Lu, H. Xia, and Z. Yi, ``A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics,'' IEEE Trans. Intell. Transp. Syst., vol. 16, no. 6, pp. 30333049, Dec. 2015.

[5] M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, ``Wearable 2.0: Enable human-cloud integration in next generation healthcare system,'' IEEE Commun., vol. 55, no. 1, pp. 5461, Jan. 2017.

[6] M. Chen, Y. Ma, J. Song, C. Lai, and B. Hu, ``Smart clothing: Con- necting human with clouds and big data for sustainable health monitor- ing,'' ACM/Springer Mobile Netw. Appl., vol. 21, no. 5, pp. 825845, 2016.

[7] M. Chen, P. Zhou, and G. Fortino, ``Emotion communi- cation system,'' IEEE Access, vol. 5, pp. 326337, 2017, doi: 10.1109/ACCESS.2016.2641480.

[8] M. Qiu and E. H.-M. Sha, ``Cost minimization while satisfying hard/soft timing constraints for heterogeneous embedded systems,'' ACM Trans. Design Autom. Electron. Syst., vol. 14, no. 2, p. 25, 2009.

[9] J. Wang, M. Qiu, and B. Guo, ``Enabling real-time information service on telehealth system over cloud-based big data platform,'' J. Syst. Archit., vol. 72, pp. 6979, Jan. 2017.

[10] D. W. Bates, S. Saria, L. Ohno-Machado, A. Shah, and G. Escobar, ``Big data in health care: Using analytics to identify and manage high-risk and high-cost patients,'' Health Affairs, vol. 33, no. 7, pp. 11231131, 2014.

[11] L. Qiu, K. Gai, and M. Qiu, ``Optimal big data sharing approach for tele- health in cloud computing,'' in Proc. IEEE Int. Conf. Smart Cloud (Smart- Cloud), Nov. 2016, pp. 184189.

[12] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, ``Health- CPS: Healthcare cyber-physical system assisted by cloud and big data,'' IEEE Syst. J., vol. 11, no. 1, pp. 8895, Mar. 2017.

[13] K. Lin, J. Luo, L. Hu, M. S. Hossain, and A. Ghoneim, ``Localization based on social big data analysis in the vehicular networks,'' IEEE Trans. Ind. Informat., to be published, doi: 10.1109/TII.2016.2641467.

[14] K. Lin, M. Chen, J. Deng, M. M. Hassan, and G. Fortino, ``Enhanced ngerprinting and trajectory prediction for iot localization in smart build- ings,'' IEEE Trans. Autom. Sci. Eng., vol. 13, no. 3, pp. 12941307, Jul. 2016.

[15] D. Oliver, F. Daly, F. C. Martin, and M. E. McMurdo, ``Risk factors and risk assessment tools for falls in hospital in-patients: A systematic review,'' Age Ageing, vol. 33, no. 2, pp. 122130, 2004.

[16] S. Marcoon, A. M. Chang, B. Lee, R. Salhi, and J. E. Hollander, ``Heart score to further risk stratify patients with low TIMI scores,'' Critical Pathways Cardiol., vol. 12, no. 1, pp. 15, 2013.