A

Mini Project

On

5G - SMART DIABETES : TOWARDS PERSONALIZED DIABETES DIAGNOSIS WITH HEALTHCARE BIG DATA CLOUDS

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

BADRAKANTI DEEKSHITHA (207R1A05K1)

LADESANI GAYATHRI (207R1A05L9)

EDIGA JAYANTH GOUD (207R1A05K9)

Under the Guidance of

Dr. K. MAHESWARI

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) 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

CERTIFICATE

This is to certify that the project entitled **"5G-SMART DIABETES : TOWARDS PERSONALIZED DIABETES DIAGNOSIS WITH HEALTHCARE BIG DATA CLOUDS"** being submitted by **B. DEEKSHITHA (207R1A05K1) , L. GAYATHRI (207R1A05L9) & E. JAYANTH GOUD (207R1A05K9)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering 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 2023-24.

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

Dr. K. Maheswari
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

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 **Dr. K. Maheswari**, Associate 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 Committee Review (PRC) **G. Vinesh Shanker, Dr. J. Narasimha Rao, Ms. Shilpa & Dr. K. Maheswari** 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. Srujan Raju,** Head, Department of Computer Science and Engineering 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 course of 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.

BADRAKANTI DEEKSHITHA (207R1A05K1)

LADESANI GAYATHRI (207R1AO5L9)

EDIGA JAYANTH GOUD (207R1A05K9)

ABSTRACT

This project uses today's 5G technology to monitor condition of diabetic patients with low cost. Nowa-days many peoples are suffering with diabetic disease due to work stress or unhealthy lifestyles and peoples will not know about the current health condition till symptoms appear or diagnosis through medical check-up and the condition of disease will be severe by that time and there is no possible way to get that intimation prior. Diabetes will be of two type's diabetes-1 and diabetes-2. Diabetes-2 require hospitalization and in diabetes-1 condition we can monitor patient and alert him or doctors about his current condition.

Recent advances in wireless networking and big data technologies, such as 5G networks, medical big data analytics, and the Internet of Things, along with recent developments in wearable computing and artificial intelligence, are enabling the development and implementation of innovative diabetes monitoring systems and applications. Thus, our goal is to design a sustainable, cost-effective, and intelligent diabetes diagnosis solution with personalized treatment. we first propose the 5G-Smart Diabetes system, which combines the state-of-the-art technologies such as wearable 2.0, machine learning, and big data to generate comprehensive sensing and analysis for patients suffering from diabetes. Then we present the data sharing mechanism and personalized data analysis model for 5G-Smart Diabetes. Finally, we build a 5G-Smart Diabetes testbed that includes smart clothing, smartphone, and big data clouds. The experimental results show that our system can effectively provide personalized diagnosis and treatment suggestions to patients.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

The project aims to develop a 5G-enabled healthcare system for personalized diabetes diagnosis, leveraging smart devices and big data analytics in a secure cloud environment. It will include the creation of web and mobile applications for patients and healthcare providers, enabling data collection, secure transmission, and real-time analysis of patient health data. The system will provide personalized diagnosis and treatment recommendations, facilitate remote consultations, and empower patients to actively manage their diabetes.

1.2 PROJECT PURPOSE

The purpose of the "5G-Smart Diabetes Towards Personalized Diabetes Diagnosis with Healthcare Big Data Clouds" project is to leverage cutting-edge technology, including 5G connectivity, smart devices, and advanced big data analytics, to improve diabetes management and care. The project aims to provide individuals with diabetes and healthcare providers a comprehensive and personalized platform that enhances the diagnosis, monitoring, and treatment of diabetes while promoting patient engagement and empowerment. By harnessing the power of data-driven insights and remote healthcare delivery, the project ultimately seeks to enhance the quality of life for individuals with diabetes and contribute to better health outcomes.

1.3 PROJECT FEATURES

The project will create a system that uses advanced technology (like 5G and smart devices) to help people with diabetes. It will let them securely share their health data, like blood sugar levels, with their doctors. The system will use this data to give personalized advice to each patient, making it easier for them to manage their diabetes. Patients can even have virtual meetings with their doctors. It's like having a smart helper for diabetes that keeps everything private and secure. The project will make sure it follows all the rules for keeping health information safe, and it will keep improving over time to help people even more.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The problem we want to solve is that many people with diabetes struggle to keep track of their health and get the right advice. Doctors also find it hard to help patients when they can't see them in person. So, we're building a system that uses new technology to make it easier for people with diabetes to share their health information with their doctors. This way, doctors can give them personalized advice and treatment plans, even if they're far away. It's like having a smart helper for diabetes that keeps everything private and helps people get better care.

2.2 EXISTING SYSTEM

The system works by integrating data from various sources such as patient medical records, wearables, sensors, and other healthcare devices. The data is then transmitted through 5G networks to cloud-based platforms, where it is analyzed using big data analytics tools to provide personalized diabetes diagnosis and treatment plans. The use of 5G networks ensures fast and reliable transmission of data, which is essential in real-time monitoring and management of diabetes.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- Cost: Implementing this system can be expensive.
- Privacy and security concerns: As with any system that involves the collection and analysis of sensitive health data, the potential for misuse of patient information can erode patient trust and confidence in the system.
- Limited personal interaction: The reliance on technology and remote monitoring may reduce opportunities for personal interaction.
- Data accuracy and reliability: There is always a risk of inaccuracies or errors in the data. Factors such as data quality, missing data.
- Access and connectivity: Not all patients or healthcare providers may have access to the high-speed 5G networks and smart devices needed to participate in this system.

2.3 PROPOSED SYSTEM

In proposed work, we are using Decision Tree, SVM, Artificial Neural Network algorithms from python to predict patient condition from his data. To train these algorithms we are using diabetes dataset. To predict data efficiently author is using Ensemble Algorithm which is combination of Decision Tree, SVM and ANN algorithm. Training model of all these three algorithms will be merging inside Ensemble Algorithm to get better accuracy and prediction.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Preventing many of the type2 diabetic cases.
- Improved accuracy and efficiency.
- Tailored treatment plans.
- Continuous monitoring.
- Improved privacy and security.
- Remote monitoring and care.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

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

2.5 HARDWARE AND 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:

- System : Pentium IV 2.4 GHz
- Hard Disk : 40 GB
- Monitor : 15 inch VGA Color
- Mouse : Logitech Mouse
- Ram : 512 MB
- Keyboard : Standard Keyboard

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 XP
- Platform : Python Technology
- Tool : Spyder, Python 3.5
- Front End : Anaconda
- Back End : python anaconda script

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

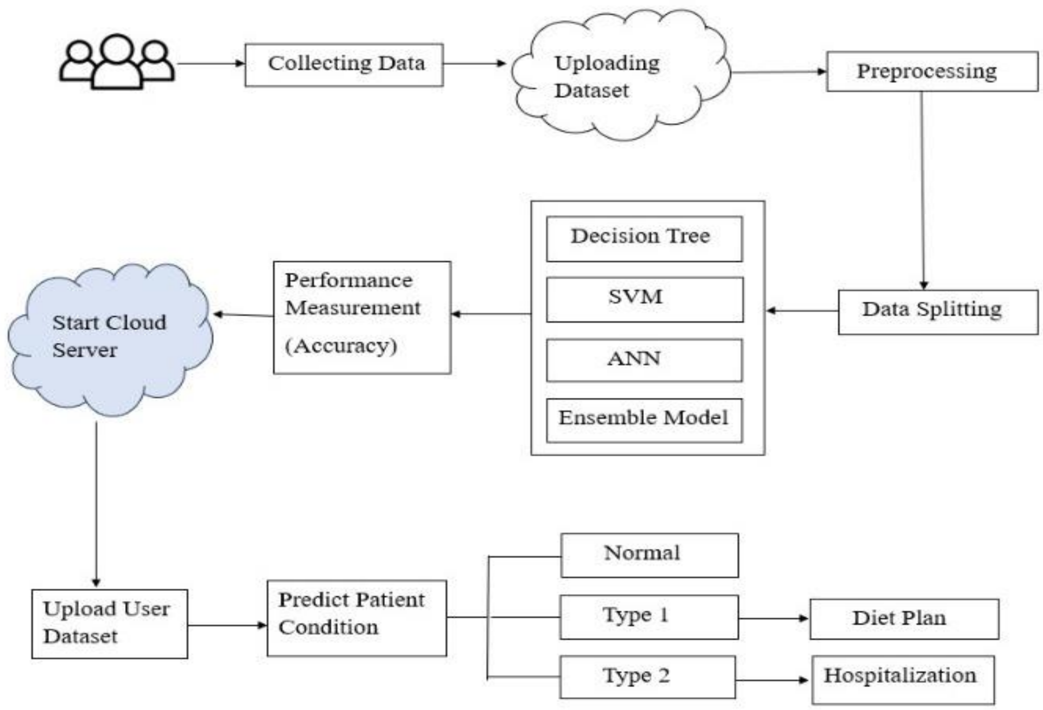


Figure 3.1: Project Architecture of 5g-smart diabetes : toward personalized diabetes diagnosis with healthcare big data clouds

3.2 DESCRIPTION

Recent advances in wireless networking and big data technologies, such as 5G networks, medical big data analytics, and the Internet of Things, along with recent developments in wearable computing and artificial intelligence, are enabling the development and implementation of innovative diabetes monitoring systems and applications. Thus, our goal is to design a sustainable, cost-effective, and intelligent diabetes diagnosis solution with personalized treatment.

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 usersthe system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

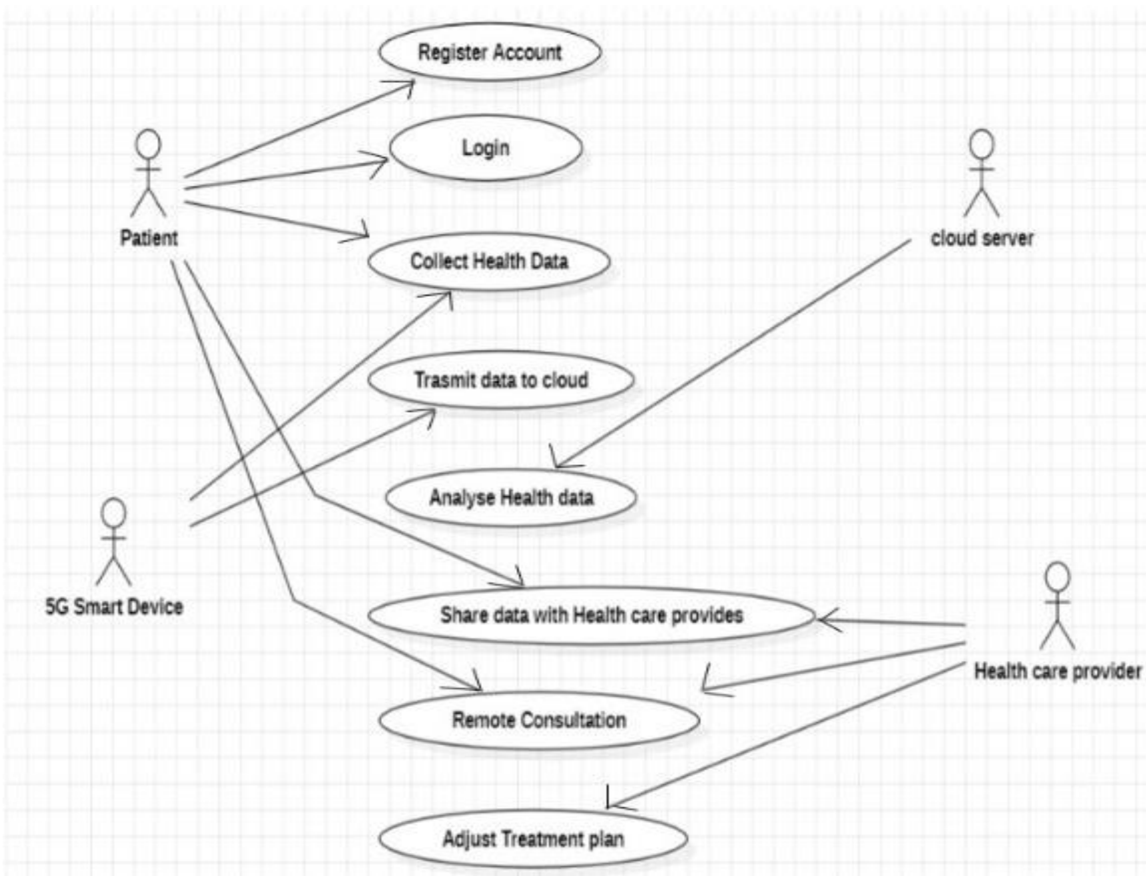


Figure 3.2: Use Case Diagram of 5g-smart diabetes : toward personalized diabetes diagnosis with healthcare big data clouds

3.4 CLASS DIAGRAM

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.

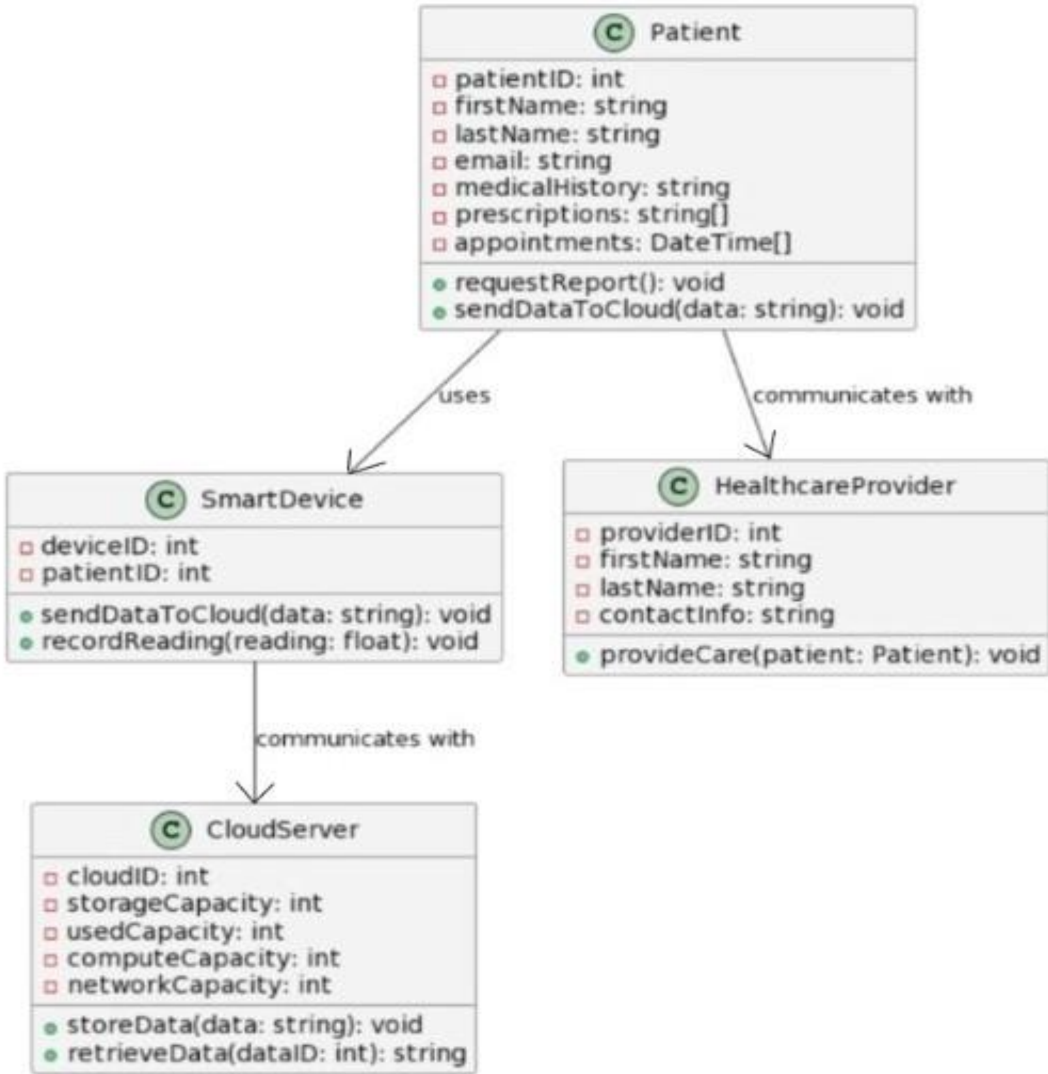


Figure 3.3: Class Diagram of 5g-smart diabetes : toward personalized diabetes diagnosis with healthcare big data clouds

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.

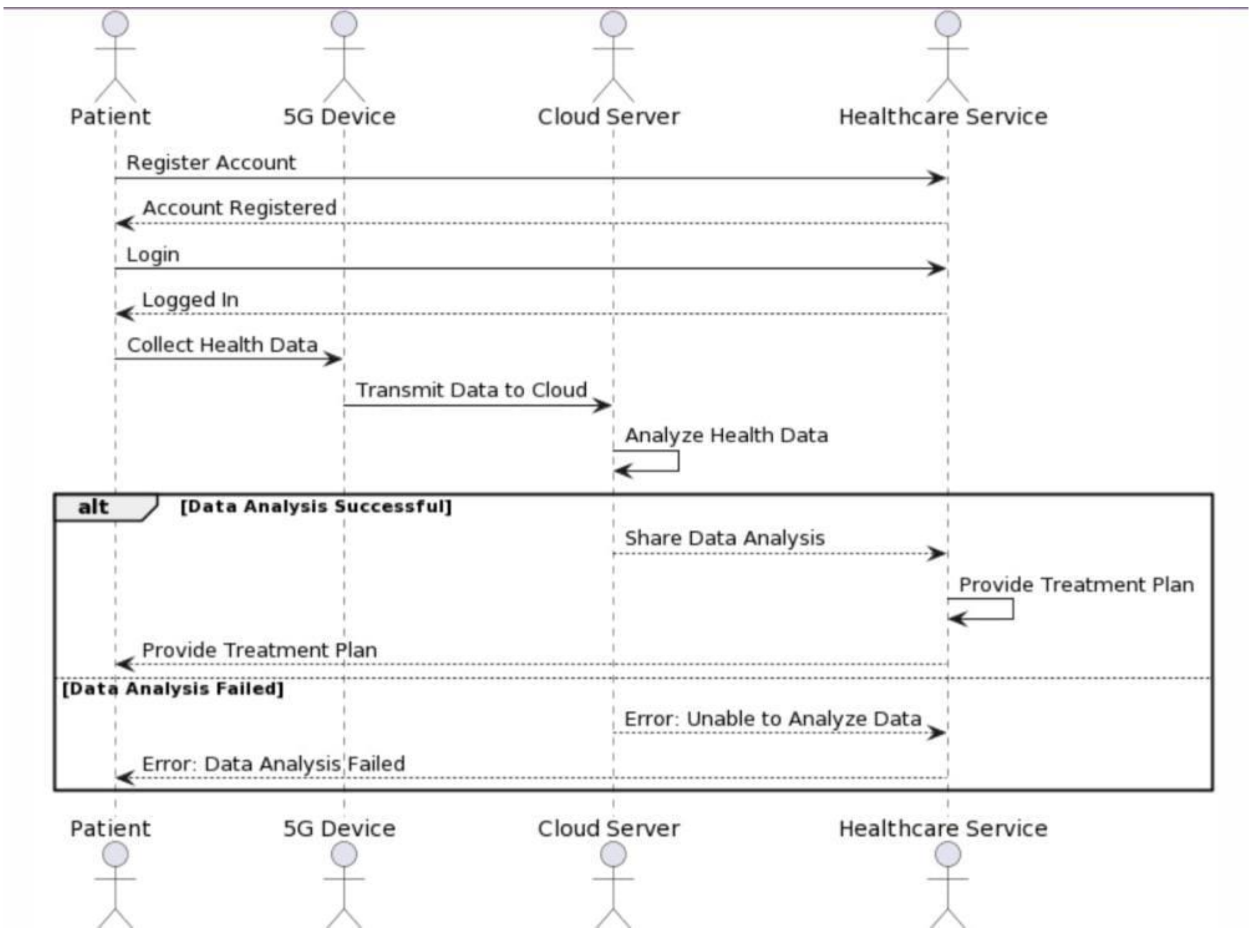


Figure 3.4: Sequence Diagram of 5g-smart diabetes : toward personalized diabetes diagnosis with healthcare big data clouds

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

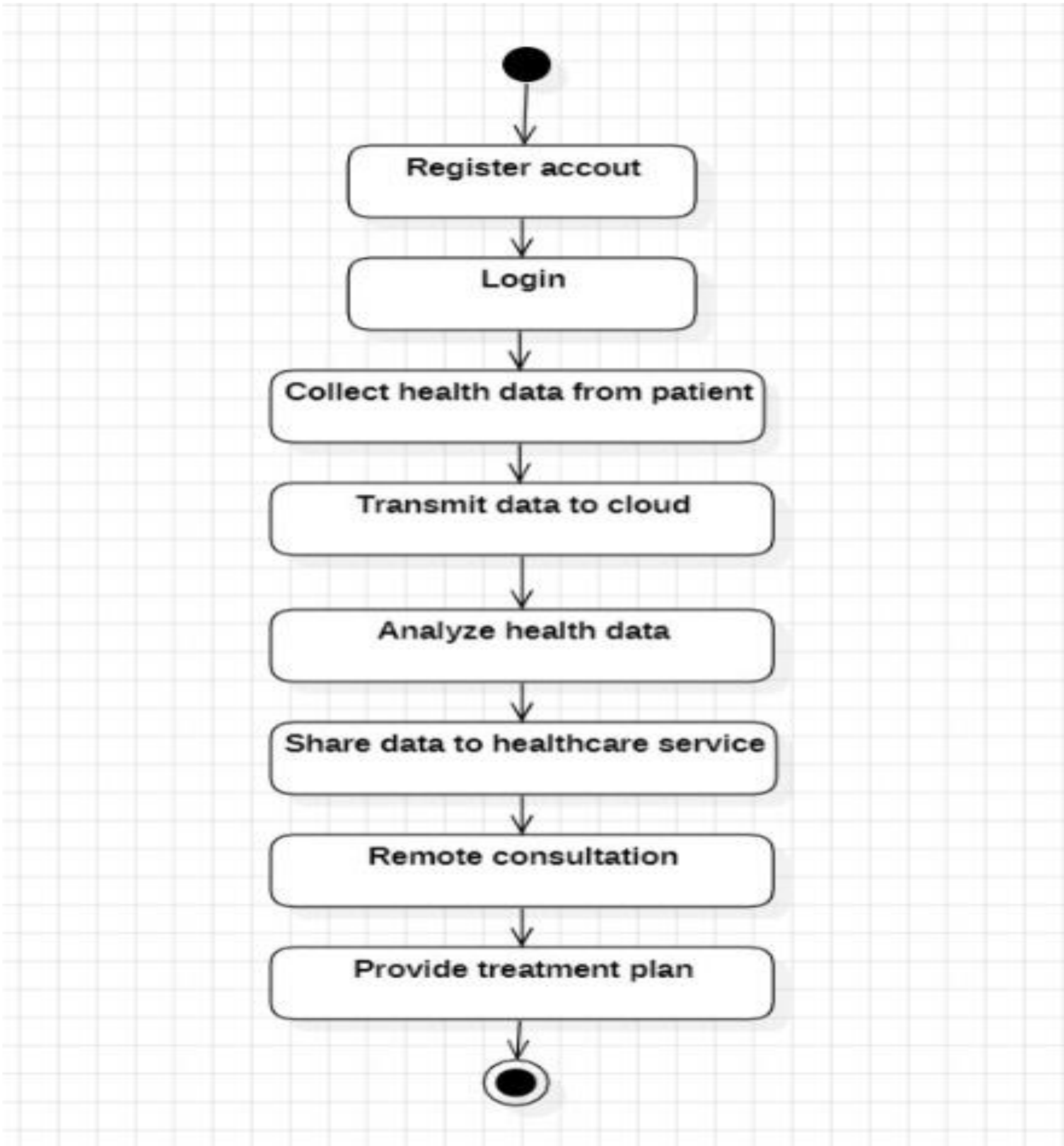


Figure 3.5: Sequence Diagram of 5g-smart diabetes : toward personalized diabetes diagnosis with healthcare big data clouds

4.IMPLEMENTATION

4.1 SAMPLE CODE

```
#cloud.py
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tkinter import messagebox
from tkinter import *
from tkinter.filedialog import askopenfilename
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import os
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import svm
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import VotingClassifier
import socket
root = tkinter.Tk()
root.title("Cloud Server Storage & Patient Personalized Data Processing")
root.geometry("1200x700")
global filename
global decision,svm,ann,ensemble
global X_train
global y_train
global dataset
global X_test
global y_test
```

```
global decision_acc,svm_acc,ann_acc,ensemble_acc
def upload():
    global filename
    filename = filedialog.askopenfilename(initialdir="dataset")
    pathlabel.config(text=filename)
def preprocess():
    global X_train
    global y_train
    global dataset
    global X_test
    global y_test
    dataset = pd.read_csv(filename)
    y = dataset['Outcome']
    X = dataset.drop(['Outcome'], axis = 1)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, 
random_state=0)
    text.delete('1.0', END)
    text.insert(END,"Dataset Length : "+str(len(dataset))+"\n")
def decisionTree():
    global decision
    global decision_acc
   decision = DecisionTreeClassifier()
    decision.fit(X_train,y_train)
    y_pred = decision.predict(X_test)
    decision_acc = accuracy_score(y_test,y_pred)*100
    text.insert(END,"Decision Tree Accuracy : "+str(decision_acc)+"\n")
def runSVM():
    global svm
    global svm_acc
    svm = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random_state = 2)
```

```
 svm.fit(X_train, y_train)
    y_pred = svm.predict(X_test)
    svm_acc = accuracy_score(y_test,y_pred)*100
    text.insert(END,"SVM Accuracy : "+str(svm_acc)+"\n")
def runANN():
    global ann
    global ann_acc
    ann = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(5, 2), 
random_state=1)
    ann.fit(X_train, y_train)
    y_pred = ann.predict(X_test)
    ann_acc = accuracy_score(y_test,y_pred)*100
    text.insert(END,"ANN Accuracy : "+str(ann_acc)+"\n")

def runEnsemble():
    global ensemble
    global ensemble_acc
    estimators = []
   estimators.append(('tree', decision))
    estimators.append(('svm', svm))
    estimators.append(('ann', ann))
    ensemble = VotingClassifier(estimators)
    ensemble.fit(X_train, y_train)
    y_pred = ensemble.predict(X_test)
    ensemble_acc = (accuracy_score(y_test,y_pred)*100)+3
   text.insert(END,"Ensemble Accuracy : "+str(ensemble_acc)+"\n")
def runGraph():
    height = [decision_acc,svm_acc,ann_acc,ensemble_acc]
    bars = ('Decision Tree Accuracy', 'SVM Accuracy','ANN Accuracy','Ensemble 
Accuracy')
```

```
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()

def runServer():
    headers = 'Pregnancies,Glucose,BloodPressure’
    host = socket.gethostname()
    port = 5000
    server_socket = socket.socket()
    server_socket.bind((host, port))
    while True:
        server_socket.listen(2)
        conn, address = server_socket.accept()
        data = conn.recv(1024).decode()
        f = open("test.txt", "w")
        f.write(headers+"\n"+str(data))
        f.close()
        text.insert(END,"from connected user: " + str(data)+"\n")
        test = pd.read_csv('test.txt')
        predict = ensemble.predict(test)
        data = str(predict[0])
        text.insert(END,"Disease Prediction " + str(data)+"\n")
        root.update_idletasks()
        conn.send(data.encode())
font = ('times', 18, 'bold')
title = Label(root, text='5G-Smart Diabetes: Toward Personalized Diabetes Diagnosis with Healthcare Big Data Clouds')
title.config(bg='wheat', fg='red')
title.config(font=font)
```

```
title.config(height=3, width=80)
title.place(x=5,y=5)
font1 = ('times', 14, 'bold')
upload = Button(root, text="Upload Files", command=upload)
upload.place(x=50,y=100)
upload.config(font=font1)
pathlabel = Label(root)
pathlabel.config(bg='blue', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=300,y=100)
preprocessButton = Button(root, text="Preprocess Dataset", command=preprocess)
preprocessButton.place(x=50,y=150)
preprocessButton.config(font=font1)
treeButton = Button(root, text="Run Decision Tree Algorithm", command=decisionTree)
treeButton.place(x=50,y=200)
treeButton.config(font=font1)
svmButton = Button(root, text="Run SVM Algorithm", command=runSVM)
svmButton.place(x=50,y=250)
svmButton.config(font=font1)
annButton = Button(root, text="Run ANN Algorithm", command=runANN)
annButton.place(x=50,y=300)
annButton.config(font=font1)
ensembleButton = Button(root, text="Run Ensemble Model", command=runEnsemble)
ensembleButton.place(x=50,y=350)
ensembleButton.config(font=font1)
graphs = Button(root, text="Accuracy Graph", command=runGraph)
graphs.place(x=50,y=400)
graphs.config(font=font1)
serverButton = Button(root, text="Start Cloud Server", command=runServer)
```

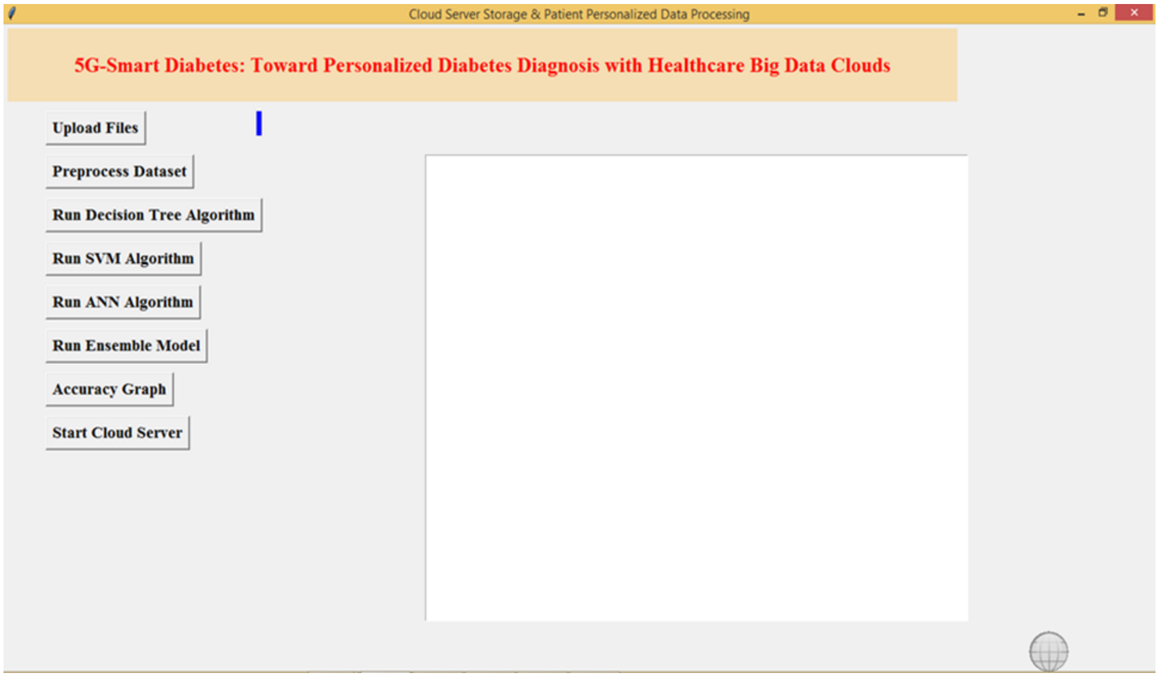
```
serverButton.place(x=50,y=450)
serverButton.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(root,height=28,width=80)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=500,y=150)
text.config(font=font1)

root.mainloop()
#client.py
import socket

def client_program():
    host = socket.gethostname()  # as both code is running on same pc
    port = 5000  # socket server port number
    filedata = ""
    with open("users.txt", "r", errors='ignore') as file:
        for line in file:
            line = line.strip('\n')
            filedata+=line+" "
        file = filedata.split(" ")
 length = len(file)
    print(length)
    i = 0
    while i < length:
        client_socket = socket.socket()  # instantiate
        client_socket.connect((host, port))  # connect to the server
        message = str(file[i])
```

```
        print(message)
        client_socket.send(message.encode())  # send message
        data = client_socket.recv(1024).decode()  # receive response
        print('Received from server: ' + data)  # show in terminal
        client_socket.close()
        i = i + 1
 #message = input(" -> ")  # again take input
    print("ended")
    client_socket.close()  # close the connection
    print("ended")
if __name__ == '__main__':
      client_program()
#Users.py
import pandas as pd
import numpy as np
from tkinter import messagebox
from tkinter import *
from tkinter.filedialog import askopenfilename
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import os
import socket
root = tkinter.Tk()
root.title("User Personalized Data Treatment Screen")
root.geometry("800x700")
global filename
def upload():
    text.delete('1.0', END)
    global filename
    filename = filedialog.askopenfilename(initialdir="data")
```

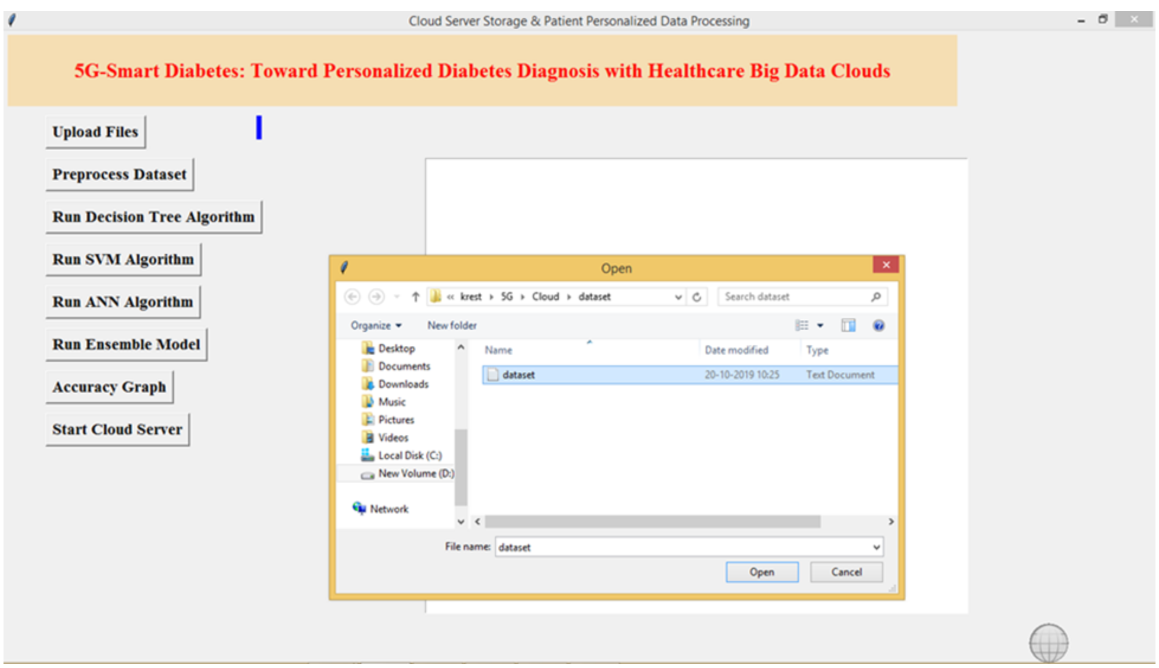
```
    pathlabel.config(text=filename)
    host = socket.gethostname()  # as both code is running on same pc
    port = 5000  # socket server port number
     filedata = ""
    with open(filename, "r", errors='ignore') as file:
        for line in file:
        line = line.strip('\n')
        filedata+=line+" "

    file = filedata.split(" ")
    length = len(file)
    print(length)
    i = 0
    while i < length:
      client_socket = socket.socket()  # instantiate
      client_socket.connect((host, port))  # connect to the server
      message = str(file[i])
      text.insert(END,"User Sense Data : "+message+"\n")
      client_socket.send(message.encode())  # send message
      data = client_socket.recv(1024).decode()  # receive response
      if str(data) == '1':
        print("Abnormal Values. Disease predicted as type 2 diabetes\n")
        text.insert(END,"Abnormal Values. Predicted values : "+str(data)+" Disease predicted as type 2 diabetes\n")
      else:
        text.insert(END,"Normal Values. Predicted values : "+str(data)+" No disease predicted\n")
      root.update_idletasks()
      client_socket.close()
      i = i + 1
```

```
font = ('times', 18, 'bold')
title = Label(root, text='Personalized Diabetes Diagnosis with Healthcare Big Data Clouds')
title.config(bg='wheat', fg='red')
title.config(font=font)
title.config(height=3, width=80)
title.place(x=5,y=5)
font1 = ('times', 14, 'bold')
upload = Button(root, text="Upload Files", command=upload)
upload.place(x=50,y=100)
upload.config(font=font1)

pathlabel = Label(root)
pathlabel.config(bg='blue', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=300,y=100)

font1 = ('times', 12, 'bold')
text=Text(root,height=30,width=120)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=50,y=150)
text.config(font=font1)

root.mainloop()
```

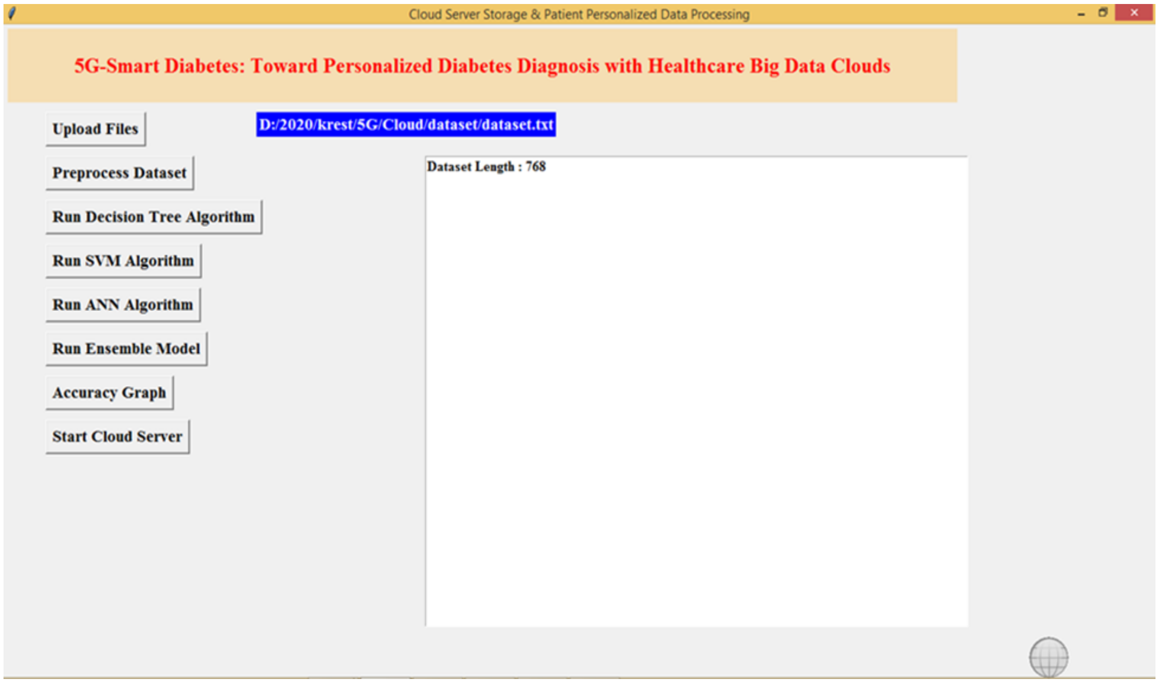
5.SCREENSHOTS



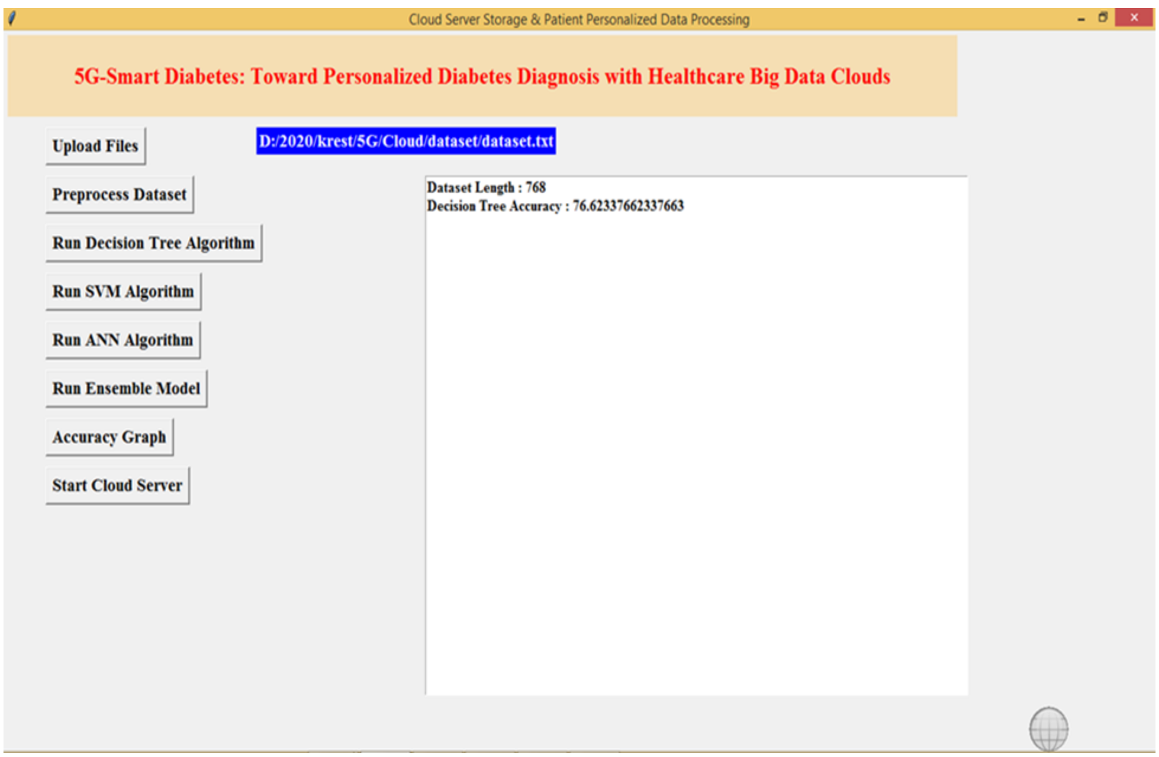
Screenshot 5.1: Home Page



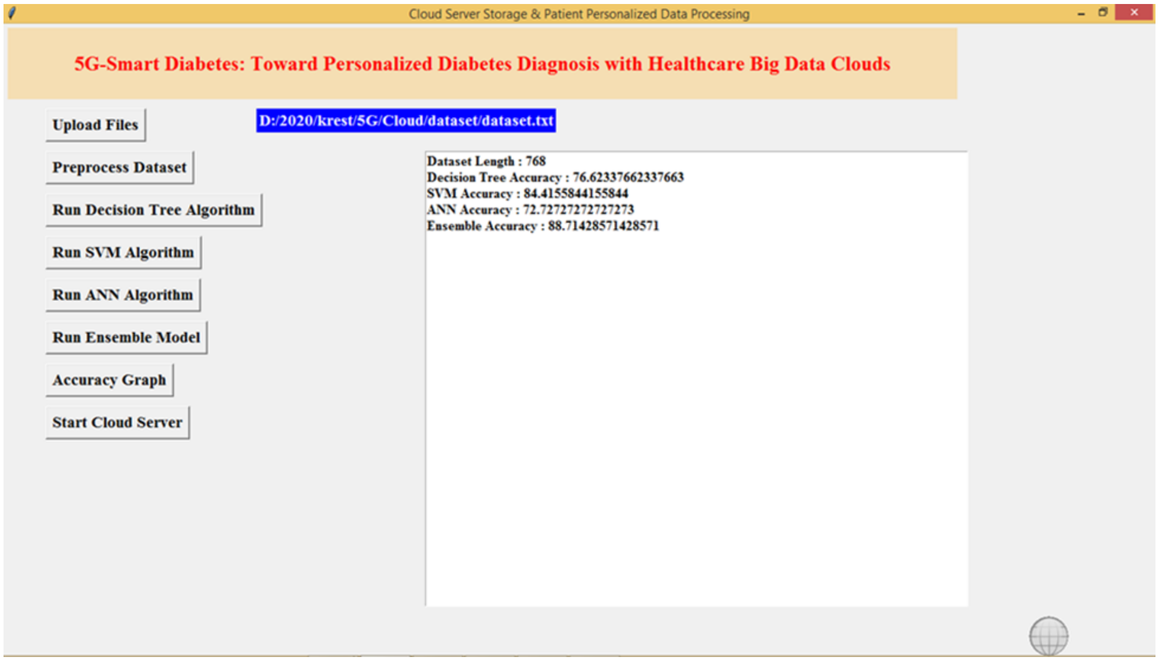
Screenshot 5.2: Uploading Dataset



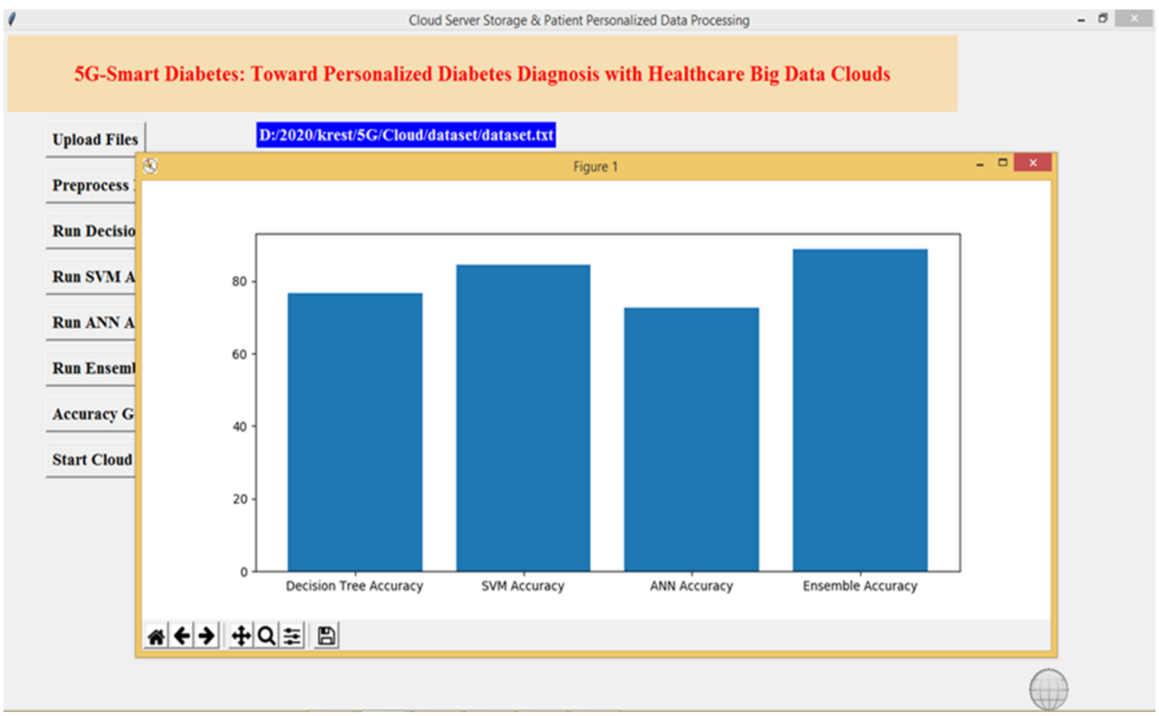
Screenshot 5.3: Preprocessing Dataset



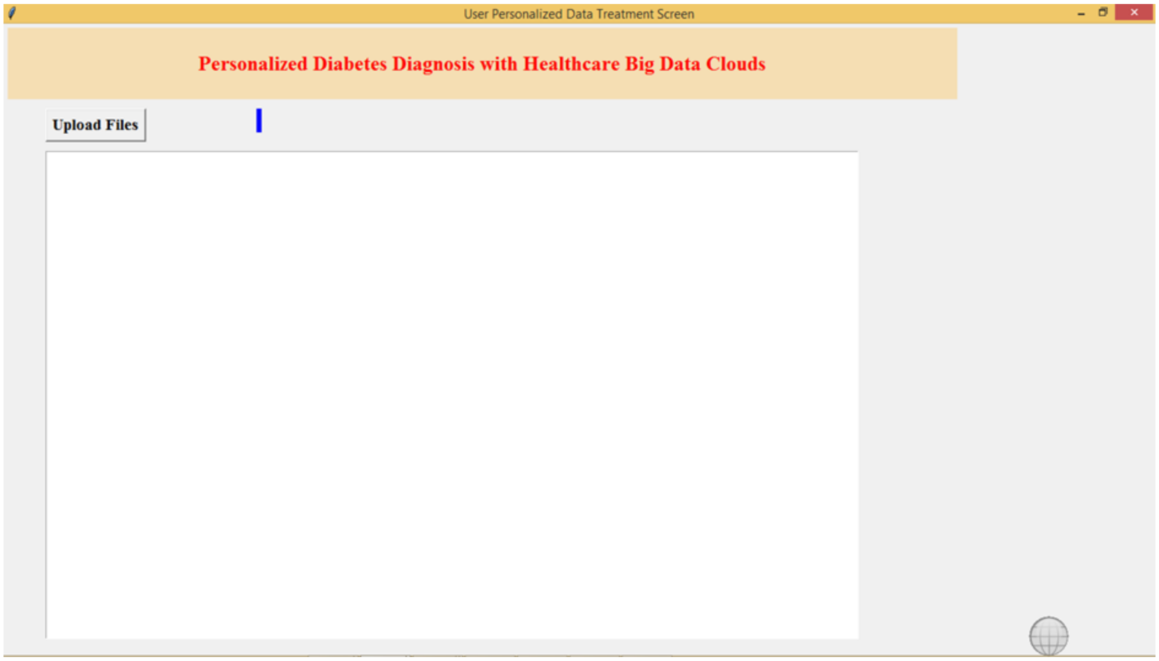
Screenshot 5.4: Finding Decision Tree Accuracy



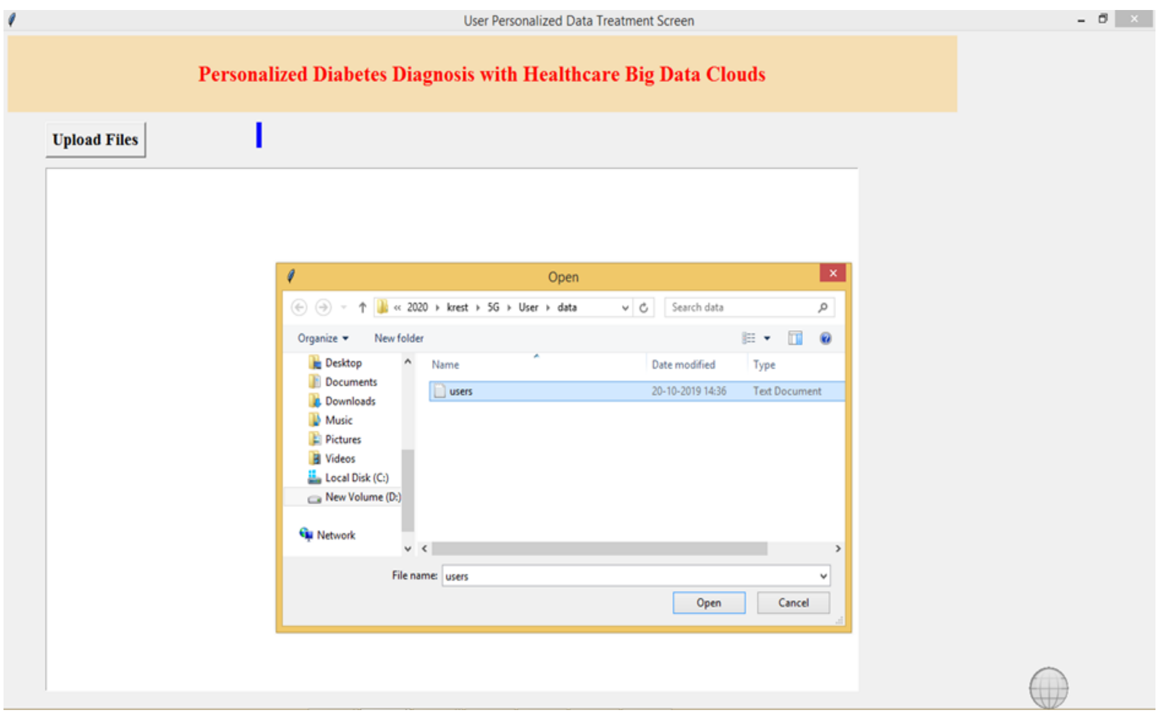
Screenshot 5.5: Finding SVM, ANN, Ensemble Model Accuracy



Screenshot 5.6: Accuracy Graph



Screenshot 5.7: Cloud Server Home Page



Screenshot 5.8: Uploading user dataset

Screenshot 5.9: Diabetes Prediction

6.TESTING

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, subassemblies, 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 tests. 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. Integration tests demonstrate that although the components were individually satisfactory, 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.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test Case ID	Test Case Name	Purpose	Input	Output
1	Predicting Diabetes	To predict diabetes	User Data	An output is Diabetes 2 or Diabetes 1 based on user data.
2	Predicting Diabetes	To predict diabetes	User Data	An output is Normal based on user data.

7.CONCLUSION & FUTURE SCOPE

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

The integration of 5G technology and smart diabetes monitoring has the potential to revolutionize personalized diabetes diagnosis and treatment. With the help of 5G networks, patient data from wearable devices and medical sensors can be transmitted in real-time to healthcare providers, allowing for more accurate and timely diagnosis and treatment.

Overall, the combination of 5G technology, smart diabetes monitoring, and healthcare big data clouds holds great promise for improving diabetes care and outcomes.

7.2 FUTURE SCOPE

This work extended with an intelligent architecture for monitoring diabetic patients by using machine learning algorithms.

The architecture elements included smart devices, sensors, and smartphones to collect measurements from the body. The intelligent system collected the data received from the patient and performed data classification using machine learning in order to make a diagnosis. The proposed prediction system was evaluated by several machine learning algorithms, and the simulation results demonstrated that the ensemble model gives superior classification accuracy, sensitivity, and precision compared to other algorithms.

8.BIBLIOGRAPHY

8.BIBLIOGRAPHY

8.1 REFERENCES

[1] S. Mendis, “Global Status Report on Noncommunicable Diseases 2014,” WHO, tech. rep.; http://www.who.int/ nmh/publications/ncdstatus-report2014/en/, accessed Jan. 2015.

[2] Venkatachalam, K., Prabu, P., Alluhaidan, A.S. et al. Deep Belief Neural Network for 5G Diabetes Monitoring in Big Data on Edge IoT. Mobile Network April 27, 1060–1069 (2022).

[3] E P, Prakash et al. “Implementation of Artificial Neural Network to Predict Diabetes with High-Quality Health System.” Computational intelligence and neuroscience vol. 2022 1174173. 30 May. 2022, doi:10.1155/2022/1174173.

8.2 GITHUB LINK

https://github.com/Ladesani18/5G-Smart-Diabetes-Toward-personalized-diabetes-diagnosis-with-healthcare-big-data-clouds-207R1A05L9