Intelligent Systems

Final Proposal

**Development of Fuzzy Expert System for the Diagnosis of Diabetes Mellitus**

**Name: Jayakody Kankanamalage Chamani Shiranthika**

**Index: 710882702**

**Research Background**

Diabetes is worldwide problem. It is one of rapidly increase disease in the world. Diabetes, referred as diabetes mellitus it is organic process in which the person has increase blood glucose (blood sugar), either because insulin origination is deficient, or body’s cells do not behave properly to insulin which is produce. Early investigate of diabetes is an important objection. In recent years, number of diabetic patients has increased mostly because of higher population and infrequent western food habits or lacking exercise. Mainly there are 2 types of diabetes of which are referred to as Type 1 diabetes and Type 2 diabetes. Type1 diabetes occurs due to historical inheritance.

* Type 1 diabetes: Insulin dependent and the human body does not produce insulin.
* Type 2 diabetes: Non-Insulin dependent and human body does not produce sufficient insulin for proper activity in the body

Researchers have undertaken several steps in detecting diabetes mellitus with applications of Neural Network, Naïve Bayes, and Support vector machine. But performance is idle of existing systems. Here I propose a quicker and more valuable technique to diagnosis of diabetes using Fuzzy Inference System. User (may be doctor or nurse) only needs to give some physical parameters. On the basis of providing information, in early stage fuzzy inference system diagnosis of diabetes whether that person is suffering or not. System gives a membership value related to the percentage of diabetes the person has and states whether he is in high risk, medium risk or in low stage of diabetes.

**Fuzzy Logic solution approach**

Fuzzy logic is a precise problem-solving methodology. It is the technique that facilitates the control of a complicated system without knowledge of its mathematical description. In fuzzy logic, a statement can assume any real value between 0 and 1, representing the degree to which an element belongs to a given set.

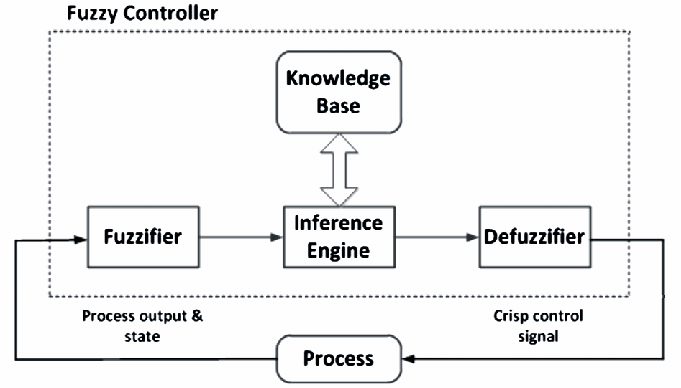


Figure 1: Fuzzy Logic controller system

Fuzzy logic is a computational paradigm that provides a mathematical tool for dealing with the uncertainty and the imprecision typical of human reasoning. A prime characteristic of fuzzy logic is its capability of expressing knowledge in a linguistic way, allowing a system to be described by simple, human-friendly rules. The fuzzy set framework has been utilized in several different approaches to modeling the diagnostic process.

1. Initialization

- Linguistic variables and terms are defined.

- Construction of the membership functions.

- Construction of the fuzzy decision tree and extraction of the fuzzy rule base.

2. Fuzzification

- This step converts crisp input data to fuzzy values using the membership functions.

3. Fuzzy inference

- Evaluate the rules in the fuzzy rule base.

- Combine the results of each rule.

Important inference systems are Mamdani, sugeno etc.

4. Defuzzification

- This step converts the output data to non-fuzzy values or real values.

Popular defuzzification methods are COA(center of area), COG (center of gravity), AI (adaptive integration), BOA (bisector of area), CDD (constraint decision defuzzification) etc.

**Proposed Methodology**

**Input and output parameters**

Input parameters

* Pregnant times (Preg)
* Glucose level (Plas)
* Diastolic BP (Dias)
* Skin Thickness (Tric)
* Serum Insulin (Ins)
* BMI (Mass)
* Pedigree (Pedi)
* Age (Age)

Output paramets

* Diabetes Mellitus (DM)

**Proposed Algorithm**

Input

Input the fuzzy set for A1, A2, A3, A4, A5, A6, A7 and A8

Output

Output the fuzzy set for DM

Method

Begin  
Step1: Input the crisp values for A1, A2, A3, A4, A5, A6, A7 and A8.  
Step 2: Set the triangular membership function for the fuzzy number with equation.  
Step 3: Built the fuzzy numbers for A1, A2, A3, A4, A5, A6, A7 and A8 for input set  
Step 3.1: Built the fuzzy number for DM for the output set.  
Step4: Fuzzy inference is executed by Mamdani’s method.  
Step 4.1: Input the rule as {Rule 1,2...k}  
Step 4.2: Matching degree of rule with OR fuzzy disjunction are calculated for fuzzy input set (A11, A12, A13, A21, A22, A23, A31, A32, A33, A41, A42, A43, A51, A52, A53, A61, A62, A63, A71, A72, A73, A81, A82, A83, DM1, DM2 and DM3).  
Step5: Defuzzify into the crisp values using centroid method  
Step6: Present the knowledge in the form of human nature language.  
End.

**Parameters of Triangular Membership Functions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fuzzy Variables** | **Representation of fuzzy** | **Fuzzy Numbers** | **Representation of fuzzy numbers** | **Fuzzy Triangular and** **Trapezoidal numbers** |
| Pregnant times | A1 | Low | A11 | [0 0.475 7.213 7.213] |
| (Preg) |  | Medium | A12 | [0.475 7.213 12.1] |
|  |  | High | A13 | [7.213 12.1 17 17] |
| Glucose level | A2 | Low | A21 | [0 88.84 152.6 152.6] |
| (Plas) |  | Medium | A22 | [88.84 152.6 175.8] |
|  |  | High | A23 | [152.6 175.8 199 199] |
| Diastolic BP | A3 | Low | A31 | [0 49.73 88.44 88.44] |
| (Dias) |  | Medium | A32 | [49.73 88.44 105.2] |
|  |  | High | A33 | [88.44 105.2 122 122] |
| Skin Thickness | A4 | Good | A41 | [0 4.584 36.49 36.49] |
| (Tric) |  | Average | A42 | [4.584 36.49 67.74] |
|  |  | Below | A43 | [36.49 67.74 99 99] |
| Serum Insulin | A5 | Low | A51 | [0 35.44 195 195] |
| (Ins) |  | Medium | A52 | [35.44 195 520.5] |
|  |  | High | A53 | [195 520.5 846 846] |
| BMI | A6 | Low | A61 | [0 24.11 39.87 39.87] |
| (Mass) |  | Medium | A62 | [24.11 39.87 53.49] |
|  |  | High | A63 | [39.87 53.49 67.1 67.1] |
| Pedigree | A7 | Low | A71 | [0 0.1405 0.8032 0.8032] |
| (Pedi) |  | Medium | A72 | [0.1405 0.8032 1.612] |
|  |  | High | A73 | [0.8032 1.612 2.42 2.42] |
| Age | A8 | Young | A81 | [0 21.48 45 45] |
| (Age) |  | Medium | A82 | [21.48 45 63] |
|  |  | Old | A83 | [45 63 81 81] |
|  |  |  |  |  |
| Diabetes Mellitus | DM | Dangerous | DM1 | [-0.4 0 0.4] |
|  |  | Medium | DM2 | [0.1 0.5 0.9] |
|  |  | Safe | DM3 | [0.6 1 1.4] |
|  |  |  |  |  |

**Proposed Fuzzy Rules**

1. If (A1 is Low) or (A2 is Low) or (A3 is Low) or (A4 is Good) or (A5 is Low) or (A6 is Low) or (A7 is Low) or (A8 is Low) then (DM is Low) (1)

2. If (A1 is Medium) or (A2 is Medium) or (A3 is Medium) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

3. If (A1 is High) or (A2 is High) or (A3 is High) or (A4 is Below Average) or (A5 is High) or (A6 is High) or (A7 is High) or (A8 is High) then (DM is High) (1)

4. If (A1 is High) or (A2 is High) or (A3 is Medium) or (A4 is Below Average) or (A5 is High) or (A6 is High) or (A7 is High) or (A8 is High) then (DM is Medium) (1)

5. If (A1 is High) or (A2 is Medium) or (A3 is High) or (A4 is Below Average) or (A5 is High) or (A6 is High) or (A7 is High) or (A8 is High) then (DM is Medium) (1)

6. If (A1 is Low) or (A2 is High) or (A3 is High) or (A4 is Good) or (A5 is Low) or (A6 is Low) or (A7 is Low) or (A8 is Low) then (DM is Low) (1)

7. If (A1 is Low) or (A2 is High) or (A3 is Medium) or (A4 is Good) or (A5 is Low) or (A6 is Low) or (A7 is Low) or (A8 is Low) then (DM is Low) (1)

8. If (A1 is Low) or (A2 is Medium) or (A3 is High) or (A4 is Good) or (A5 is Low) or (A6 is Low) or (A7 is Low) or (A8 is Low) then (DM is Low) (1)

9. If (A1 is Medium) or (A2 is High) or (A3 is Medium) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

10. If (A1 is Medium) or (A2 is Medium) or (A3 is High) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

11. If (A1 is High) or (A2 is Medium) or (A3 is Medium) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

12. If (A1 is Medium) or (A2 is High) or (A3 is Medium) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

13. If (A1 is Medium) or (A2 is Medium) or (A3 is High) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

14. If (A1 is Medium) or (A2 is Medium) or (A3 is Medium) or (A4 is Below Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

15. If (A1 is Medium) or (A2 is Medium) or (A3 is Medium) or (A4 is Average) or (A5 is High) or (A6 is Medium) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

16. If (A1 is Medium) or (A2 is Medium) or (A3 is Medium) or (A4 is Average) or (A5 is Medium) or (A6 is High) or (A7 is Medium) or (A8 is Medium) then (DM is Medium) (1)

17. If (A1 is Medium) or (A2 is Medium) or (A3 is Medium) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is High) or (A8 is Medium) then (DM is Medium) (1)

18. If (A1 is Medium) or (A2 is Medium) or (A3 is Medium) or (A4 is Average) or (A5 is Medium) or (A6 is Medium) or (A7 is Medium) or (A8 is Old) then (DM is Medium) (1)

**Implementation**

**Technologies used: -**

1. Matlab
2. Python 3 with TKinter interface
3. Python PIL (pip install Pillow)

**System Design: -**

The main program is written using python. Basic UI and code structure was implemented with python. Fuzzy if-then rules were established with the use of fisrule objects in matlab that is related to input membership function conditions to corresponding output membership functions. Thus the logics related to the fuzzy inference engine was written using matlab. Graph of the output membership function was also created using matlab. I used Fuzzy Logic Tool Box functionalities.

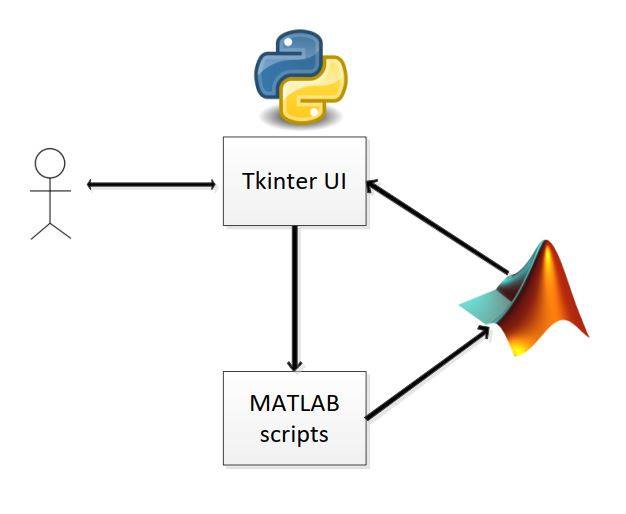
****

Figure 2: Basic System Design

**UI Design:**

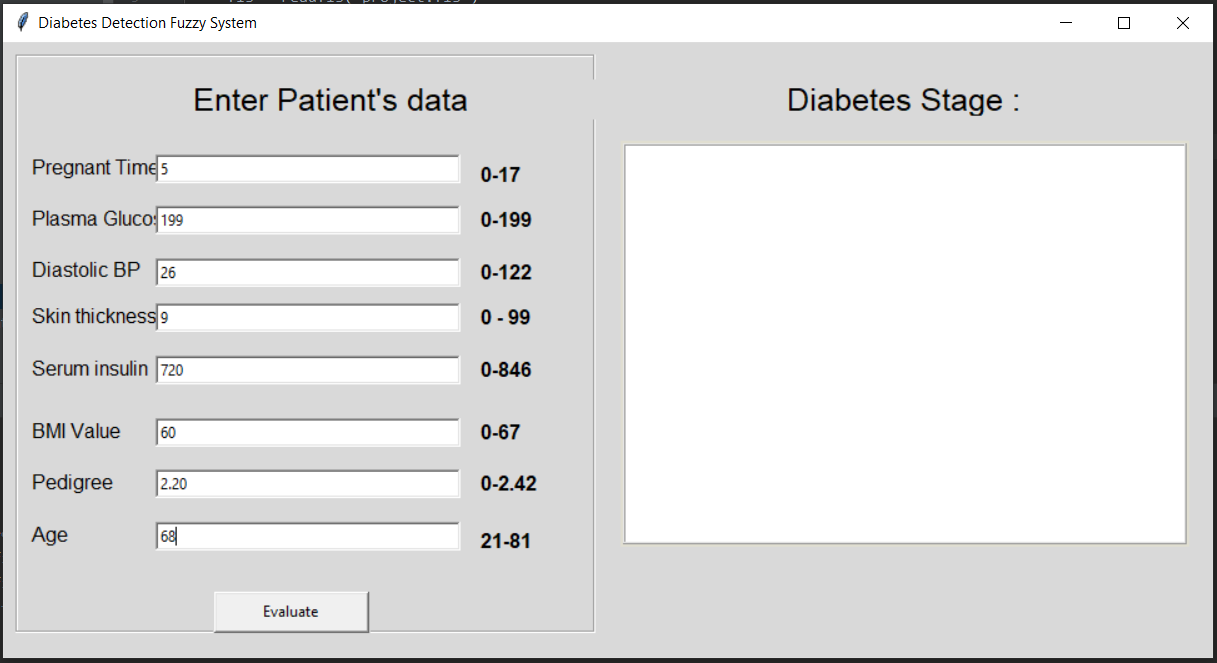


Figure 3: UI to enter patient’s data

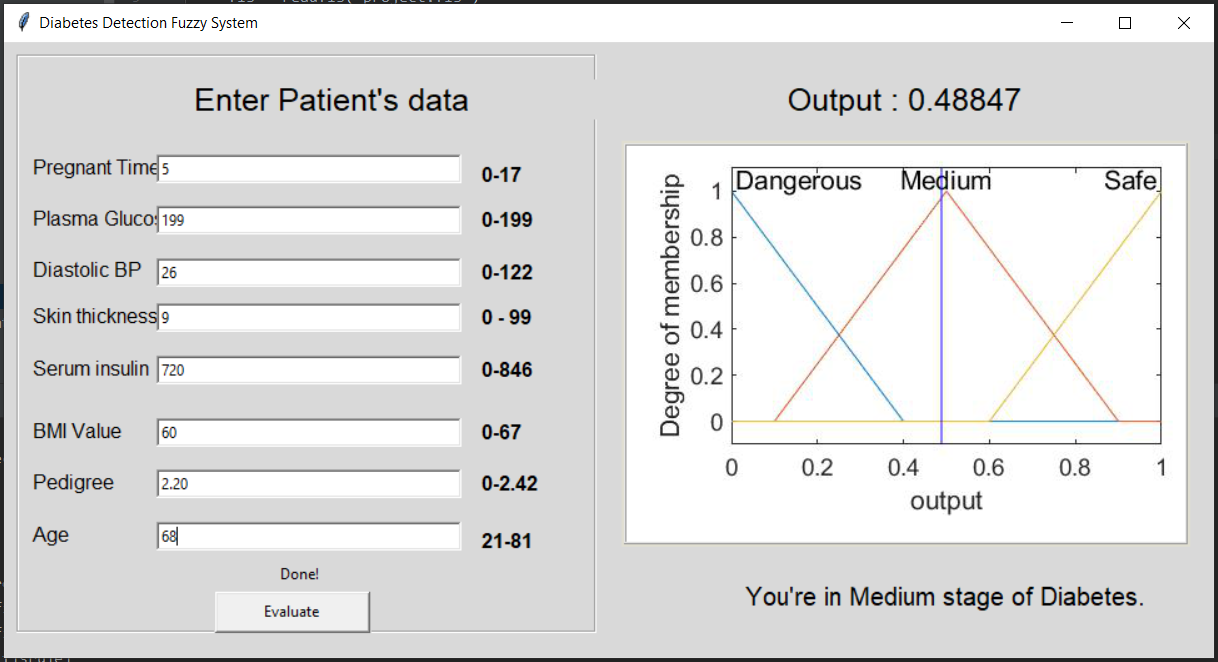


Figure 4: UI giving the Results

**Source code:**

Main logic and UI design : main.py, main\_support.py

Create output graph: createOutputGraph.m

Fuzzy rules: project.fis

Connecting fuzzy rules with main code: evalFuzzy.m

*Fuzzy inference system, project. fis code: -*

[System]  
Name='project'  
Type='mamdani'  
Version=2.0  
NumInputs=8  
NumOutputs=1  
NumRules=18  
AndMethod='min'  
OrMethod='max'  
ImpMethod='min'  
AggMethod='max'  
DefuzzMethod='centroid'  
  
[Input1]  
Name='Preg'  
Range=[0 17]  
NumMFs=3  
MF1='Low':'trapmf',[0 0.475 7.213 7.213]  
MF2='Medium':'trimf',[0.475 7.213 12.1]  
MF3='High':'trimf',[7.213 12.1 17 17]  
  
[Input2]  
Name='Plas'  
Range=[0 199]  
NumMFs=3  
MF1='Low':'trapmf',[0 88.84 152.6 152.6]  
MF2='Medium':'trimf',[88.84 152.6 175.8]  
MF3='High':'trimf',[152.6 175.8 199 199]  
  
[Input3]  
Name='Dias'  
Range=[0 122]  
NumMFs=3  
MF1='Low':'trapmf',[0 49.73 88.44 88.44]  
MF2='Medium':'trimf',[49.73 88.44 105.2]  
MF3='High':'trapmf',[88.44 105.2 122 122]  
  
[Input4]  
Name='Tric'  
Range=[0 99]  
NumMFs=3  
MF1='Good':'trapmf',[0 4.584 36.49 36.49]  
MF2='Average':'trimf',[4.584 36.49 67.74]  
MF3='Below':'trimf',[36.49 67.74 99 99]  
  
  
[Input5]  
Name='Ins'  
Range=[0 846]  
NumMFs=3  
MF1='Low':'trapmf',[0 35.44 195 195]  
MF2='Medium':'trimf',[35.44 195 520.5]  
MF3='High':'trimf',[195 520.5 846 846]  
  
  
[Input6]  
Name='Mass'  
Range=[0 67.1]  
NumMFs=3  
MF1='Low':'trapmf',[0 24.11 39.87 39.87]  
MF2='Medium':'trimf',[24.11 39.87 53.49]  
MF3='High':'trimf',[39.87 53.49 67.1 67.1]  
  
  
[Input7]  
Name='Pedi'  
Range=[0 2.42]  
NumMFs=3  
MF1='Low':'trapmf',[0 0.1405 0.8032 0.8032]  
MF2='Medium':'trimf',[0.1405 0.8032 1.612]  
MF3='High':'trimf',[0.8032 1.612 2.42 2.42]  
  
[Input8]  
Name='Age'  
Range=[0 81]  
NumMFs=3  
MF1='Young':'trapmf',[0 21.48 45 45]  
MF2='Medium':'trimf',[21.48 45 63]  
MF3='Old':'trimf',[45 63 81 81]  
  
[Output1]  
Name='output'  
Range=[0 1]  
NumMFs=3  
MF1='Dangerous':'trimf',[-0.4 0 0.4]  
MF2='Medium':'trimf',[0.1 0.5 0.9]  
MF3='Safe':'trimf',[0.6 1 1.4]  
  
  
[Rules]  
1 1 1 1 1 1 1 1, 1 (1) : 2  
2 2 2 2 2 2 2 2, 2 (1) : 2  
3 3 3 3 3 3 3 3, 3 (1) : 2  
3 3 2 3 3 3 3 3, 2 (1) : 2  
3 2 3 3 3 3 3 3, 2 (1) : 2  
1 3 3 1 1 1 1 1, 1 (1) : 2  
1 2 3 1 1 1 1 1, 1 (1) : 2  
1 2 3 1 1 1 1 1, 1 (1) : 2  
2 3 2 2 2 2 2 2, 2 (1) : 2  
2 2 3 2 2 2 2 2, 2 (1) : 2  
1 2 2 2 2 2 2 2, 2 (1) : 2  
2 3 2 2 2 2 2 2, 2 (1) : 2  
2 2 3 2 2 2 2 2, 2 (1) : 2  
2 2 2 3 2 2 2 2, 2 (1) : 2  
2 2 2 2 3 2 2 2, 2 (1) : 2  
2 2 2 2 2 3 2 2, 2 (1) : 2  
2 2 2 2 2 2 3 2, 2 (1) : 2  
2 2 2 2 2 2 2 3, 2 (1) : 2

*Main file, main. py code: -*

#! /usr/bin/env python  
  
import sys  
import matlab.engine  
eng = matlab.engine.start\_matlab()  
  
try:  
 from Tkinter import \*  
except ImportError:  
 from tkinter import \*  
  
try:  
 import ttk  
 py3 = 0  
except ImportError:  
 import tkinter.ttk as ttk  
 py3 = 1  
  
import matlab.engine  
import main\_support  
from PIL import Image, ImageTk  
import threading  
  
def vp\_start\_gui():  
 *'''Starting point when module is the main routine.'''* global val, w, root  
 root = Tk()  
 top = New\_Toplevel\_1 (root)  
 main\_support.init(root, top)  
 root.mainloop()  
  
w = None  
def create\_New\_Toplevel\_1(root, \*args, \*\*kwargs):  
 *'''Starting point when module is imported by another program.'''* global w, w\_win, rt  
 rt = root  
 w = Toplevel (root)  
 top = New\_Toplevel\_1 (w)  
 main\_support.init(w, top, \*args, \*\*kwargs)  
 return (w, top)  
  
def destroy\_New\_Toplevel\_1():  
 global w  
 w.destroy()  
 w = None  
  
class New\_Toplevel\_1:  
 def \_\_init\_\_(self, top=None):  
 *'''This class configures and populates the toplevel window.  
 top is the toplevel containing window.'''* \_bgcolor = '#d9d9d9' # X11 color: 'gray85'  
 \_fgcolor = '#000000' # X11 color: 'black'  
 \_compcolor = '#d9d9d9' # X11 color: 'gray85'  
 \_ana1color = '#d9d9d9' # X11 color: 'gray85'   
 \_ana2color = '#d9d9d9' # X11 color: 'gray85'   
 font11 = "-family Arial -size 19 -weight normal -slant roman " \  
 "-underline 0 -overstrike 0"  
 font12 = "-family Arial -size 12 -weight normal -slant roman " \  
 "-underline 0 -overstrike 0"  
 font14 = "-family Arial -size 15 -weight normal -slant roman " \  
 "-underline 0 -overstrike 0"  
 font15 = "-family Arial -size 12 -weight bold -slant roman " \  
 "-underline 0 -overstrike 0"  
 self.style = ttk.Style()  
 if sys.platform == "win32":  
 self.style.theme\_use('winnative')  
 self.style.configure('.',background=\_bgcolor)  
 self.style.configure('.',foreground=\_fgcolor)  
 self.style.configure('.',font="TkDefaultFont")  
 self.style.map('.',background=  
 [('selected', \_compcolor), ('active',\_ana2color)])  
  
 top.geometry("968x493+919+245")  
 top.title("Diabetes Detection Fuzzy System")  
 top.configure(background="#d9d9d9")  
 top.configure(highlightbackground="#b9b9b9")  
 top.configure(highlightcolor="black")  
  
 self.TFrame1 = ttk.Frame(top)  
 self.TFrame1.place(relx=0.01, rely=0.02, relheight=0.94, relwidth=0.48)  
 self.TFrame1.configure(relief=GROOVE)  
 self.TFrame1.configure(borderwidth="2")  
 self.TFrame1.configure(relief=GROOVE)  
 self.TFrame1.configure(width=465)  
  
 self.TLabel1 = ttk.Label(self.TFrame1)  
 self.TLabel1.place(relx=0.3, rely=0.04, height=32, width=350)  
 self.TLabel1.configure(background="#d9d9d9")  
 self.TLabel1.configure(foreground="#000000")  
 self.TLabel1.configure(font=font11)  
 self.TLabel1.configure(relief=FLAT)  
 self.TLabel1.configure(text='''Enter Patient's data ''')  
  
#----------------------------pregnant times----------------------------------  
 self.TLabel2 = ttk.Label(self.TFrame1)  
 self.TLabel2.place(relx=0.02, rely=0.15, height=39, width=120)  
 self.TLabel2.configure(background="#d9d9d9")  
 self.TLabel2.configure(foreground="#000000")  
 self.TLabel2.configure(font=font12)  
 self.TLabel2.configure(relief=FLAT)  
 self.TLabel2.configure(text='''Pregnant Times''')  
  
 self.TEntry\_Preg = ttk.Entry(self.TFrame1)  
 self.TEntry\_Preg.place(relx=0.24, rely=0.17, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_Preg.configure(width=246)  
 self.TEntry\_Preg.configure(takefocus="")  
 self.TEntry\_Preg.configure(cursor="ibeam")  
  
#-----------------------------------plas----------------------------------------  
 self.TLabel3 = ttk.Label(self.TFrame1)  
 self.TLabel3.place(relx=0.02, rely=0.24, height=39, width=120)  
 self.TLabel3.configure(background="#d9d9d9")  
 self.TLabel3.configure(foreground="#000000")  
 self.TLabel3.configure(font=font12)  
 self.TLabel3.configure(relief=FLAT)  
 self.TLabel3.configure(text='''Plasma Glucose Level''')  
  
 self.TEntry\_Plas = ttk.Entry(self.TFrame1)  
 self.TEntry\_Plas.place(relx=0.24, rely=0.26, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_Plas.configure(width=246)  
 self.TEntry\_Plas.configure(takefocus="")  
 self.TEntry\_Plas.configure(cursor="ibeam")  
  
#---------------------------------Diastolic bp-------------------------------  
  
 self.TLabel4 = ttk.Label(self.TFrame1)  
 self.TLabel4.place(relx=0.02, rely=0.33, height=39, width=120)  
 self.TLabel4.configure(background="#d9d9d9")  
 self.TLabel4.configure(foreground="#000000")  
 self.TLabel4.configure(font=font12)  
 self.TLabel4.configure(relief=FLAT)  
 self.TLabel4.configure(text='''Diastolic BP''')  
  
 self.TEntry\_Dias = ttk.Entry(self.TFrame1)  
 self.TEntry\_Dias.place(relx=0.24, rely=0.35, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_Dias.configure(width=246)  
 self.TEntry\_Dias.configure(takefocus="")  
 self.TEntry\_Dias.configure(cursor="ibeam")  
  
#----------------------------------------Tric----------------------------------  
  
 self.TLabel5 = ttk.Label(self.TFrame1)  
 self.TLabel5.place(relx=0.02, rely=0.41, height=39, width=120)  
 self.TLabel5.configure(background="#d9d9d9")  
 self.TLabel5.configure(foreground="#000000")  
 self.TLabel5.configure(font=font12)  
 self.TLabel5.configure(relief=FLAT)  
 self.TLabel5.configure(text='''Skin thickness''')  
  
 self.TEntry\_Tric = ttk.Entry(self.TFrame1)  
 self.TEntry\_Tric.place(relx=0.24, rely=0.43, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_Tric.configure(width=246)  
 self.TEntry\_Tric.configure(takefocus="")  
 self.TEntry\_Tric.configure(cursor="ibeam")  
  
#-----------------------------Insulin-----------------------------------------  
  
 self.TLabel6 = ttk.Label(self.TFrame1)  
 self.TLabel6.place(relx=0.02, rely=0.5, height=39, width=120)  
 self.TLabel6.configure(background="#d9d9d9")  
 self.TLabel6.configure(foreground="#000000")  
 self.TLabel6.configure(font=font12)  
 self.TLabel6.configure(relief=FLAT)  
 self.TLabel6.configure(text='''Serum insulin''')  
  
 self.TEntry\_Ins = ttk.Entry(self.TFrame1)  
 self.TEntry\_Ins.place(relx=0.24, rely=0.52, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_Ins.configure(width=246)  
 self.TEntry\_Ins.configure(takefocus="")  
 self.TEntry\_Ins.configure(cursor="ibeam")  
  
#-----------------------------BMI Value--------------------------------------  
  
 self.TLabel6 = ttk.Label(self.TFrame1)  
 self.TLabel6.place(relx=0.02, rely=0.61, height=39, width=120)  
 self.TLabel6.configure(background="#d9d9d9")  
 self.TLabel6.configure(foreground="#000000")  
 self.TLabel6.configure(font=font12)  
 self.TLabel6.configure(relief=FLAT)  
 self.TLabel6.configure(text='''BMI Value''')  
  
 self.TEntry\_bmi = ttk.Entry(self.TFrame1)  
 self.TEntry\_bmi.place(relx=0.24, rely=0.63, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_bmi.configure(width=246)  
 self.TEntry\_bmi.configure(takefocus="")  
 self.TEntry\_bmi.configure(cursor="ibeam")  
  
#-----------------------------Pedigree------------------------------------  
  
 self.TLabel6 = ttk.Label(self.TFrame1)  
 self.TLabel6.place(relx=0.02, rely=0.70, height=39, width=120)  
 self.TLabel6.configure(background="#d9d9d9")  
 self.TLabel6.configure(foreground="#000000")  
 self.TLabel6.configure(font=font12)  
 self.TLabel6.configure(relief=FLAT)  
 self.TLabel6.configure(text='''Pedigree''')  
  
 self.TEntry\_Pedigree = ttk.Entry(self.TFrame1)  
 self.TEntry\_Pedigree.place(relx=0.24, rely=0.72, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_Pedigree.configure(width=246)  
 self.TEntry\_Pedigree.configure(takefocus="")  
 self.TEntry\_Pedigree.configure(cursor="ibeam")  
  
#-----------------------------Age-----------------------------------------  
  
 self.TLabel6 = ttk.Label(self.TFrame1)  
 self.TLabel6.place(relx=0.02, rely=0.79, height=39, width=120)  
 self.TLabel6.configure(background="#d9d9d9")  
 self.TLabel6.configure(foreground="#000000")  
 self.TLabel6.configure(font=font12)  
 self.TLabel6.configure(relief=FLAT)  
 self.TLabel6.configure(text='''Age''')  
  
 self.TEntry\_Age = ttk.Entry(self.TFrame1)  
 self.TEntry\_Age.place(relx=0.24, rely=0.81, relheight=0.05  
 , relwidth=0.53)  
 self.TEntry\_Age.configure(width=246)  
 self.TEntry\_Age.configure(takefocus="")  
 self.TEntry\_Age.configure(cursor="ibeam")  
# -----------------------------------------------------------------------  
  
 self.TButton\_eval = ttk.Button(self.TFrame1)  
 self.TButton\_eval.place(relx=0.34, rely=0.93, height=35, width=126)  
 self.TButton\_eval.configure(takefocus="")  
 self.TButton\_eval.configure(text='''Evaluate''')  
 self.TButton\_eval.bind('<Button-1>',lambda e:main\_support.TButton\_eval\_onClick(e))  
  
  
 self.TLabel\_Indicator = ttk.Label(self.TFrame1)  
 self.TLabel\_Indicator.place(relx=0.29, rely=0.88, height=19, width=184)  
 self.TLabel\_Indicator.configure(background="#d9d9d9")  
 self.TLabel\_Indicator.configure(foreground="#000000")  
 self.TLabel\_Indicator.configure(relief=FLAT)  
 self.TLabel\_Indicator.configure(anchor=CENTER)  
  
 self.TLabel7 = ttk.Label(self.TFrame1)  
 self.TLabel7.place(relx=0.8, rely=0.18, height=19, width=36)  
 self.TLabel7.configure(background="#d9d9d9")  
 self.TLabel7.configure(foreground="#000000")  
 self.TLabel7.configure(font=font15)  
 self.TLabel7.configure(relief=FLAT)  
 self.TLabel7.configure(anchor=W)  
 self.TLabel7.configure(text='''0-17''')  
 self.TLabel7.configure(width=36)  
  
 self.TLabel8 = ttk.Label(self.TFrame1)  
 self.TLabel8.place(relx=0.8, rely=0.26, height=19, width=46)  
 self.TLabel8.configure(background="#d9d9d9")  
 self.TLabel8.configure(foreground="#000000")  
 self.TLabel8.configure(font=font15)  
 self.TLabel8.configure(relief=FLAT)  
 self.TLabel8.configure(anchor=W)  
 self.TLabel8.configure(text='''0-199''')  
 self.TLabel8.configure(width=46)  
  
 self.TLabel9 = ttk.Label(self.TFrame1)  
 self.TLabel9.place(relx=0.8, rely=0.35, height=19, width=66)  
 self.TLabel9.configure(background="#d9d9d9")  
 self.TLabel9.configure(foreground="#000000")  
 self.TLabel9.configure(font=font15)  
 self.TLabel9.configure(relief=FLAT)  
 self.TLabel9.configure(anchor=W)  
 self.TLabel9.configure(text='''0-122''')  
 self.TLabel9.configure(width=66)  
  
 self.TLabel11 = ttk.Label(self.TFrame1)  
 self.TLabel11.place(relx=0.8, rely=0.43, height=19, width=76)  
 self.TLabel11.configure(background="#d9d9d9")  
 self.TLabel11.configure(foreground="#000000")  
 self.TLabel11.configure(font=font15)  
 self.TLabel11.configure(relief=FLAT)  
 self.TLabel11.configure(anchor=W)  
 self.TLabel11.configure(text='''0 - 99''')  
 self.TLabel11.configure(width=76)  
  
 self.TLabel10 = ttk.Label(self.TFrame1)  
 self.TLabel10.place(relx=0.8, rely=0.52, height=19, width=56)  
 self.TLabel10.configure(background="#d9d9d9")  
 self.TLabel10.configure(foreground="#000000")  
 self.TLabel10.configure(font=font15)  
 self.TLabel10.configure(relief=FLAT)  
 self.TLabel10.configure(anchor=W)  
 self.TLabel10.configure(text='''0-846''')  
 self.TLabel10.configure(width=56)  
  
 self.TLabel10 = ttk.Label(self.TFrame1)  
 self.TLabel10.place(relx=0.8, rely=0.63, height=19, width=56)  
 self.TLabel10.configure(background="#d9d9d9")  
 self.TLabel10.configure(foreground="#000000")  
 self.TLabel10.configure(font=font15)  
 self.TLabel10.configure(relief=FLAT)  
 self.TLabel10.configure(anchor=W)  
 self.TLabel10.configure(text='''0-67''')  
 self.TLabel10.configure(width=56)  
  
 self.TLabel10 = ttk.Label(self.TFrame1)  
 self.TLabel10.place(relx=0.8, rely=0.72, height=19, width=56)  
 self.TLabel10.configure(background="#d9d9d9")  
 self.TLabel10.configure(foreground="#000000")  
 self.TLabel10.configure(font=font15)  
 self.TLabel10.configure(relief=FLAT)  
 self.TLabel10.configure(anchor=W)  
 self.TLabel10.configure(text='''0-2.42''')  
 self.TLabel10.configure(width=56)  
  
  
 self.TLabel10 = ttk.Label(self.TFrame1)  
 self.TLabel10.place(relx=0.8, rely=0.82, height=19, width=56)  
 self.TLabel10.configure(background="#d9d9d9")  
 self.TLabel10.configure(foreground="#000000")  
 self.TLabel10.configure(font=font15)  
 self.TLabel10.configure(relief=FLAT)  
 self.TLabel10.configure(anchor=W)  
 self.TLabel10.configure(text='''21-81''')  
 self.TLabel10.configure(width=56)  
  
 self.TLabel\_Output = ttk.Label(top)  
 self.TLabel\_Output.place(relx=0.52, rely=0.06, height=29, width=436)  
 self.TLabel\_Output.configure(background="#d9d9d9")  
 self.TLabel\_Output.configure(foreground="#000000")  
 self.TLabel\_Output.configure(font=font11)  
 self.TLabel\_Output.configure(relief=FLAT)  
 self.TLabel\_Output.configure(anchor=CENTER)  
 self.TLabel\_Output.configure(text='''Diabetes Stage :''')  
 self.TLabel\_Output.configure(width=436)  
  
 self.Canvas\_Graph = Canvas(top)  
 self.Canvas\_Graph.place(relx=0.51, rely=0.16, relheight=0.66  
 , relwidth=0.47)  
 self.Canvas\_Graph.configure(background="white")  
 self.Canvas\_Graph.configure(borderwidth="2")  
 self.Canvas\_Graph.configure(highlightbackground="#e0ded1")  
 self.Canvas\_Graph.configure(highlightcolor="black")  
 self.Canvas\_Graph.configure(insertbackground="black")  
 self.Canvas\_Graph.configure(relief=RIDGE)  
 self.Canvas\_Graph.configure(selectbackground="#cac8bc")  
 self.Canvas\_Graph.configure(selectforeground="black")  
 self.Canvas\_Graph.configure(width=456)  
  
 self.TLabel\_OutputText = ttk.Label(top)  
 self.TLabel\_OutputText.place(relx=0.52, rely=0.87, height=29, width=500)  
 self.TLabel\_OutputText.configure(background="#d9d9d9")  
 self.TLabel\_OutputText.configure(foreground="#000000")  
 self.TLabel\_OutputText.configure(font=font14)  
 self.TLabel\_OutputText.configure(relief=FLAT)  
 self.TLabel\_OutputText.configure(anchor=CENTER)  
 self.TLabel\_OutputText.configure(width=500)  
  
 widgets = [self.TEntry\_Preg, self.TEntry\_Plas, self.TEntry\_Dias, self.TEntry\_Tric, self.TEntry\_Ins, self.TEntry\_bmi,self.TEntry\_Pedigree,self.TEntry\_Age]  
 self.TButton\_eval.bind('<Button-1>',lambda e:self.TButton\_eval\_onClick(e, widgets))  
  
 self.TEntry\_Preg.insert(END, '7')  
 self.TEntry\_Plas.insert(END, '140')  
 self.TEntry\_Dias.insert(END, '80')  
 self.TEntry\_Tric.insert(END, '50')  
 self.TEntry\_Ins.insert(END, '500')  
 self.TEntry\_bmi.insert(END, '40')  
 self.TEntry\_Pedigree.insert(END, '1.40')  
 self.TEntry\_Age.insert(END, '50')  
  
 # start matlab engine  
 self.matlabeng = matlab.engine.start\_matlab()  
  
  
 def TButton\_eval\_onClick(self, p1, widgets):  
  
 self.TLabel\_Indicator['text'] = "Processing from MATLAB..."  
 self.TButton\_eval.state(['disabled'])  
  
 # run in multithread so our UI won't freeze  
 args = [float(x.get()) for x in widgets]  
 t = threading.Thread(target=self.doMATLABProcessing, args=[args])  
 t.daemon = True  
 t.start()  
  
 def doMATLABProcessing(self, data):  
  
 # contacting MATLAB using its API  
 val = self.matlabeng.evalFuzzy(\*data, nargout=1)  
 self.matlabeng.createOutputGraph(val, nargout=0)  
 self.tk\_img = ImageTk.PhotoImage(file='output.jpg')  
  
 # displaying MATLAB output to the interface  
 self.Canvas\_Graph.create\_image(250, 150, image=self.tk\_img)  
 self.TLabel\_Output['text'] = "Output : %.5f" % val  
 self.TLabel\_Indicator['text'] = "Done!"  
 self.outputOutputMsg(val)  
  
 # enable the button back  
 self.TButton\_eval.state(['!disabled'])  
  
 def outputOutputMsg(self, val):  
 cond = []  
 if val >= 0.1 and val <= 0.9: cond += ['medium']  
 if val >= 0 and val <= 0.4: cond += ['dangerous']  
 if val >= 0.6 and val <= 1.0: cond += ['safe']  
 if len(cond) == 1:  
 if val <= 0.1:  
 self.TLabel\_OutputText['text'] = "You're in Dangerous stage of Diabetes.".format(\*cond)  
 if val >= 0.4 and val <= 0.6:  
 self.TLabel\_OutputText['text'] = "You're in Medium stage of Diabetes.".format(\*cond)  
 if val >= 0.7:  
 self.TLabel\_OutputText['text'] = "You're in Safe stage of Diabetes.".format(\*cond)  
  
 else:  
 if val >= 0.1 and val <= 0.4:  
 self.TLabel\_OutputText['text'] = "You're in Medium and Dangerous stage of Diabetes.".format(\*cond)  
 if val >= 0.6 and val <= 0.7:  
 self.TLabel\_OutputText['text'] = "You're in Medium and Safe stage of Diabetes.".format(\*cond)

if \_\_name\_\_ == '\_\_main\_\_':  
 vp\_start\_gui()

*Main code support file, main\_support.py code: -*

#! /usr/bin/env python  
import sys  
  
try:  
 from Tkinter import \*  
except ImportError:  
 from tkinter import \*  
  
try:  
 import ttk  
 py3 = 0  
except ImportError:  
 import tkinter.ttk as ttk  
 py3 = 1  
  
def TButton\_eval\_onClick(p1):  
 print('main\_support.TButton\_eval\_onClick')  
 sys.stdout.flush()  
  
def init(top, gui, \*args, \*\*kwargs):  
 global w, top\_level, root  
 w = gui  
 top\_level = top  
 root = top  
  
def destroy\_window():  
 # Function which closes the window.  
 global top\_level  
 top\_level.destroy()  
 top\_level = None  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 import main  
 main.vp\_start\_gui()

*Create output graph file, createOutputGraph.m code:-*

function [] = createOutputGraph(x)  
 fis = readfis('project.fis')  
 h = figure;  
 set(h, 'visible', 'off');  
 set(h,'PaperUnits','inches','PaperPosition',[0 0 3 2])  
 set(h,'DefaultTextFontSize', 14)  
 plotmf(fis, 'output', 1)  
 line([x, x], ylim, 'Color','b')  
 saveas(h, 'output.jpg')  
end

*Connecting fuzzy rules with main code file, evalFuzzy.m code:-*

function val = evalFuzzy(Preg, Plas, Dias, Tric, Ins, Mass, Pedi, Age)  
 fis = readfis('project.fis')  
 val = evalfis([Preg, Plas, Dias, Tric, Ins, Mass, Pedi, Age], fis)  
end

**Accuracy Checking**

The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. I used the confusion matrix in order to derive the accuracy of the fuzzy prediction system.

Thus Accuracy can be stated as,

**Accuracy = TN + TP**

**TN + TP + FN + FP**

I formulated the terms as follows.

True Positive (TP): Sick people correctly identified as sick

True Positive (TP): Sick people correctly identified as sick

True Negative (TN): Healthy people correctly identified as healthy

True Negative (TN): Healthy people correctly identified as healthy

Results achieved are as follows.

|  |  |  |
| --- | --- | --- |
|  | Predicted : No | Predicted : Yes |
| Actual : No | TN = 22 | FP = 2 |
| Actual : Yes | FN = 2 | TP = 24 |

Here I used pima Indians diabetes database to evaluate the results from the fuzzy prediction system.

**System Accuracy**: 22 + 24 = 92 %

22 + 24 + 2 + 2

This is how the result evaluation with the pima Indian database was taken place.

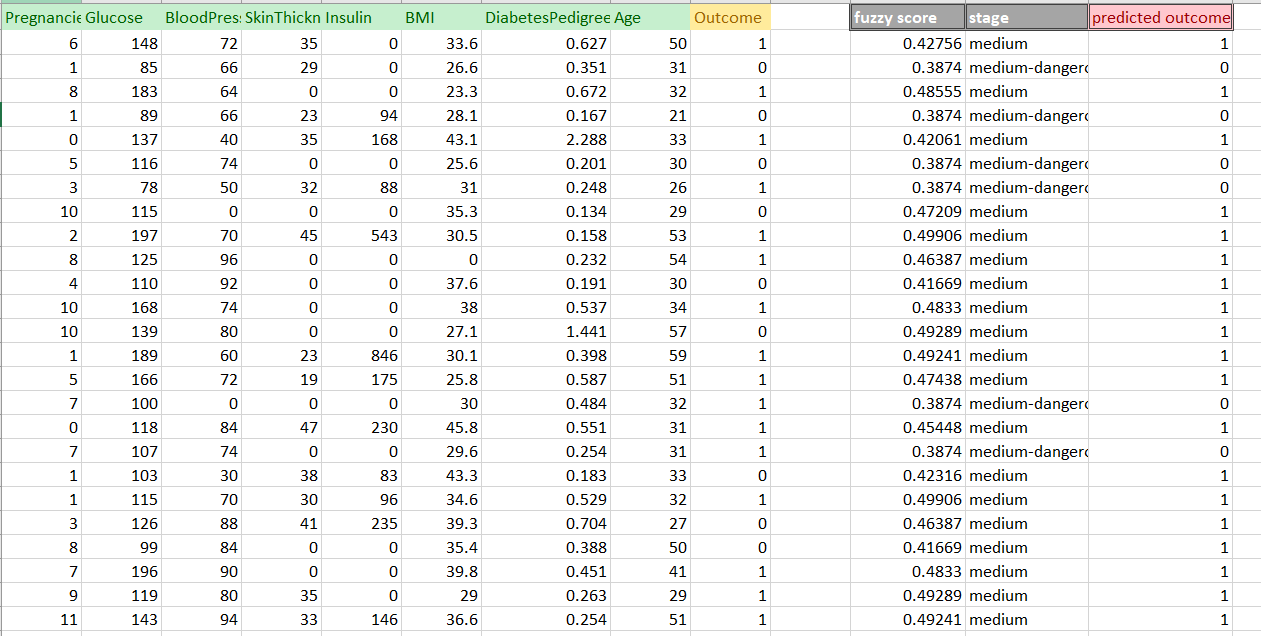


Figure 5: Evaluation of Results

**References:**

[1] H. Prajapati, A. Jain, and S. K. Pal, “An Enhance Expert System for Diagnosis of Diabetes using Fuzzy Rules over PIMA Dataset,” *International Journal of Advance Engineering and Research Development*, vol. 4, no. 9, p. 6.

[2] A. A. Abdullah, N. S. Fadil, and W. Khairunizam, “Development of Fuzzy Expert System for Diagnosis of Diabetes,” in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, Kuching, 2018, pp. 1–8, doi: 10.1109/ICASSDA.2018.8477635.

[3] M. Benamina, B. Atmani, and S. Benbelkacem, “Diabetes Diagnosis by Case-Based Reasoning and Fuzzy Logic,” *IJIMAI*, vol. 5, no. 3, p. 72, 2018, doi: 10.9781/ijimai.2018.02.001.

[4] N. Chandgude and S. Pawar, “Diagnosis of diabetes using Fuzzy inference System,” in *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, Pune, India, 2016, pp. 1–6, doi: 10.1109/ICCUBEA.2016.7860001.

[5] Z. Niswati, F. A. Mustika, and A. Paramita, “Fuzzy logic implementation for diagnosis of *Diabetes Mellitus* disease at Puskesmas in East Jakarta,” *J. Phys.: Conf. Ser.*, vol. 1114, p. 012107, Nov. 2018, doi: 10.1088/1742-6596/1114/1/012107.

[6] Department of CSE & IT, ITM University, Gurgaon-122017, India, V. Jain, and S. Raheja, “Improving the Prediction Rate of Diabetes using Fuzzy Expert System,” *IJITCS*, vol. 7, no. 10, pp. 84–91, Sep. 2015, doi: 10.5815/ijitcs.2015.10.10.