Python scikit-fuzzy: developing a fuzzy expert system for diabetes diagnosis

Tajul Rosli Razak[1], Ahmad Zia Ul-Saufie[2], Mohamad Hanis Yusoff[3], Mohammad Hafiz Ismail[3], Shukor Sanim Mohd Fauzi[3], Nurul Ain Mohd Zaki[4]

[1]School of Computing Sciences, College of Computing, Informatics and Media, Universiti Teknologi MARA, Shah Alam, Malaysia
[2]School of Mathematical Sciences, College of Computing, Informatics and Media, Universiti Teknologi MARA, Shah Alam, Malaysia
[3]Computing Sciences Studies, College of Computing, Informatics and Media, Universiti Teknologi MARA, Perlis Branch, Arau, Malaysia
[4]Surveying Science and Geomatics Studies, College of Built Environment, Universiti Teknologi MARA, Perlis Branch, Arau, Malaysia

Article Info

Article history:

Received Jan 5, 2023
Revised Mar 15, 2023
Accepted Mar 27, 2023

Keywords:

Diabetes diagnosis
Fuzzy expert system
Medical application
Python programming

ABSTRACT

Nowadays, improvements in diabetes detection that provide patients with vital information are needed. This is due to the fact that Diabetes mellitus has generated a worldwide epidemic that costs society and people. Also, patients tend to misread symptoms, and clinicians who collect insufficient data may produce erroneous outcomes. Therefore, this study aims to demonstrate that a programme that integrates expert advice such as decisions, recommendations, or solutions is an excellent method for reducing the incidence of diabetes. Specifically, this study intends to implement a fuzzy expert system that can detect and report the early stages of diabetes as a viable approach. Furthermore, since this programme is available to everyone, people may easily self-diagnose themselves if they have a blood glucose monitoring device. However, developing the fuzzy expert system for real-world situations, such as diabetes patients, using any programming tools is not straightforward. Therefore, this study will provide a comprehensive approach to constructing a fuzzy expert system using the popular programming language Python.

Corresponding Author:

Tajul Rosli Razak
School of Computing Sciences, College of Computing, Informatics and Media, Universiti Teknologi MARA
40450 Shah Alam, Selangor, Malaysia
Email: tajulrosli@uitm.edu.my

1. INTRODUCTION

Diabetes is a chronic condition that impairs the body's ability to metabolise food [1]. Diabetes is typically caused by either insufficient insulin production or improper insulin use. Three significant forms of diabetes exist type 1, 2, and gestational diabetes (diabetes while pregnant) [2]. Diabetes type 1 can develop at any age but is more common in children and teenagers. Type 2 diabetes is the most prevalent form of diabetes found in older persons. However, it is becoming more commonplace in children, adolescents, and young adults due to rising rates of obesity, inactivity, and poor nutrition. Gestational diabetes is a form of diabetes that can develop in women who do not already have diabetes during pregnancy. It occurs when the body is unable to produce sufficient insulin during pregnancy.

Malaysia has the highest rate of diabetes in the Western Pacific and is among the highest in the world, with an annual cost of over USD 600 million [3]. The prevalence of diabetes increased by 68.3% from

11.2% in 2011 to 18.3% in 2019 [4]. Consequently, it is crucial to create a diabetic diagnosis system to solve this problem.

Fuzzy logic has proven to be a brilliant and excellent importance tool for innovative medical systems [5], [6]. It provides an instant, simple method and precise assessment analysis of medical information to the system without specialists or medical professionals. These systems' outcomes depend on the knowledge base developed into them and obtained from field experts. Several medical diagnosis systems have already been designed and used in treatments and diagnoses. Therefore, it is worth introducing the fuzzy expert system for diagnosing diabetes.

However, it is challenging to model fuzzy expert systems for real-world applications since unambiguous design methodologies have yet to be fully established [7]. This paper will present a method for modelling the fuzzy expert system in the medical application that is diabetes diagnosis using the Python scikit-fuzzy. The contribution of this paper is as follows: i) Incorporate the expert's knowledge and ideas into the fuzzy expert system; ii) Modelling the task in the fuzzy expert system using the Python scikit-fuzzy; iii) Obtains the fuzzy rules from experts; and iv) Produce self-diagnosis diabetes application.

2. BACKGROUND

2.1. Diabetes disease

Diabetes mellitus (DM) is a metabolic condition with elevated blood glucose levels. This occurs when insulin production is insufficient or nonexistent, needing insulin injections [8]. The World Health Organization classifies diabetes mellitus into three categories. However, only three types of diabetes are expected: type 1 and type 2 diabetes, as well as gestational diabetes. Low insulin levels can develop classes requiring frequent insulin therapy [9]. Diabetes type 2 is more prevalent in adults aged 45 and older. However, many adolescents and young adults are also affected. Type 2 is the same as type 1. Insulin-resistant cells are those that have responded to insulin.

Diabetes is connected with chronic damage, organ failure, and organ injuries, including kidneys, nerves, the eyes, the heart, and blood vessels [10]. Severe hyperglycemia is characterised by polydipsia, polyuria, frequent intense hunger, weight loss, and poor vision. Chronic hyperglycemia can also be brought on by developmental failure or exposure to sickness. Frequently, untreated diabetes appears as two life-threatening symptoms: non-ketotic hyperosmolar syndrome and diabetic ketoacidosis. To avoid or prevent diabetes, we must perform regular screening early because this disease can be fatal if detected and treated late. Therefore, in this study, we offer a system for early-stage diabetes diagnosis.

2.2. Fuzzy logic systems

Zadeh introduced fuzzy logic (1965) [11]. An actual number between 0 and 1 describes the fuzzy set's degree of membership, where 0 represents "false", and 1 represents "true". Figure 1–stages of a fuzzy expert system–illustrates the process involved in the framework of fuzzy expert systems.

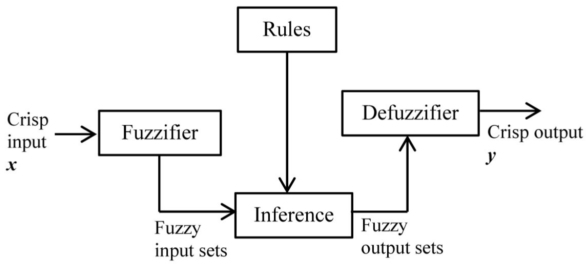

Figure 1. Fuzzy logic system. Adapted from [12]

Generally, fuzzy logic converts input to output using if-then statements, often known as rules. This rule will then be assessed with variables [12]. In addition, fuzzy logic is a mathematical function that uses fuzzy rules to convert input data to a scalar value. Concurrently, numerous outputs are viewed as a distinct multi-input set or a technique of multiple logic derived from fuzzy set theory frequently applied when a similar conclusion suffices [13].

In the medical field, fuzzy systems are widely used. As discussed in [14], fuzzy logic is a method of computation that focuses on ”degrees of truth” rather than the traditional ”true or false” (1 or 0) for modern machines relying on Boolean logic. Fuzzy logic is intended to resolve problems by considering all available data. As a result, the right decisions can be taken depending on your suggestions. Furthermore, any ambiguity can be easily handled using fuzzy logic [12]. The pillar of human speech is also the base of fuzzy logic. This is because fuzzy logic is founded on the mechanisms used for qualitative definition in daily lessons. Therefore, fuzzy systems seem to be a key to explainable artificial intelligence (XAI) [15]. However, since medical applications have too many complex variables, developing a fuzzy logic system in practice is challenging [5].

2.3. Python programming language

Python is a widely-used computer language with dynamic semantics [16]. Python syntax is simple to learn and comprehend, reducing software maintenance expenses. There are numerous modules and packages for Python. Its library is also freely distributable and available in source or binary form for all leading platforms. Many free resources are available on the internet, making it easy to study. Python also uses English grammar that is relatively straightforward to learn and comprehend. Python is well-liked by programmers due to its speed and absence of segmentation issues; it also increases productivity [17]. The Python integrated development environment (IDE) depicted in Figure 2 is Python’s editor utilised most frequently by programmers.

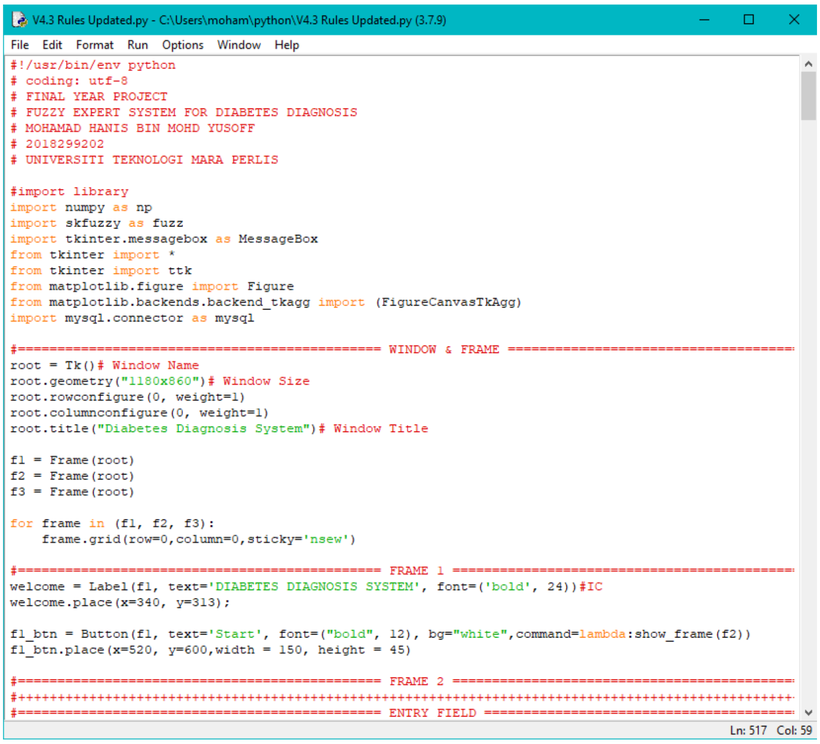

Figure 2. Python IDE interface

3. PROBLEM STATEMENT

The decision-making system produces inaccurate and imprecise outcomes due to patient misinterpretation of symptoms, physicians’ insufficient examinations, and inadequate appraisal of accessible information [18]. Consequently, it leads to less accurate diagnosis, the weighing or prioritisation of less significant analyses, and the failure to recognise the significance of some challenges [19]. In addition, they observed that time constraints and a constantly evolving standard of care led to physician errors and delayed clinical decisions. This strategy can be improved with the assistance of an expert system to the point where a decision-making system produces accurate and precise results [20]. Therefore, the development of this technology makes it possible for anyone interested to determine whether or not they have diabetes.

Python, a popular and open-source programming language [21], is advantageous for developing a system to diagnose diabetes since it is both current and extensible. However, it is difficult because it entails a medical application involving considerable decision-making uncertainty. In medical applications, the study proposes to employ the fuzzy expert system, which is adept at handling uncertainty and imprecise data. However, developing a fuzzy expert system for diabetes detection using Python remains problematic due to the lack of a consistent development guideline for real-world applications in Python [22].

4. FUZZY EXPERT SYSTEM APPROACH

This study presents an approach for developing and implementing the fuzzy expert system, which relies heavily on Python's Tkinter module [23]. Also, the rule-based fuzzy expert system function was created with the Python module scikit-fuzzy (skfuzzy) [24], which implements fuzzy logic techniques. Matplotlib, an advanced Python tool for generating static, animated, and interactive visualisations, was used to create the automatically generated graph for this method. In addition, this method includes rules that the fuzzy expert system will utilise to make decisions based on user input and respond appropriately.

Figure 3 illustrates the element involved in the fuzzy expert system that applies in this study. The framework includes the user interface, domain expert and database, as in Figure 3. Some of the elements are described in the following subsections.

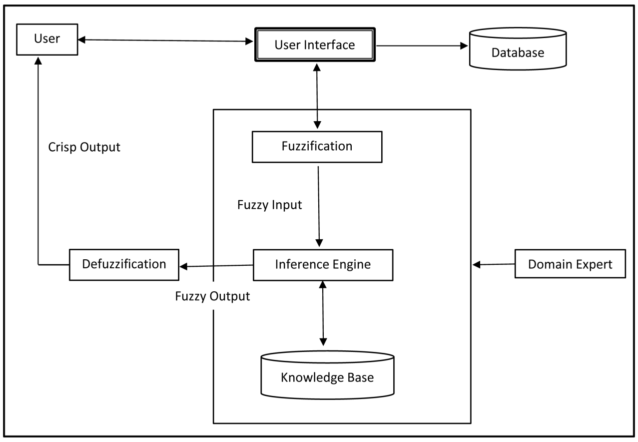

Figure 3. Fuzzy expert system framework

4.1. Fuzzification

This approach uses seven attributes (symptoms): random blood glucose (RBG), fasting blood glucose (FBG), dizziness, abrupt weight loss, blurred vision, often hunger and fatigue. Yes or No answers are provided for dizziness, short weight loss, blurred vision, often hunger and fatigue. These replies will be converted to 0 if no symptoms are present and 1 if symptoms are present. They will be calculated using the user's random and fasting blood glucose measurements to define the type of diabetes, whether it is expected, pre-diabetes, or diabetes. It should be noted that the other five attributes mentioned above (dizziness, abrupt weight loss, blurred vision, often hunger, and fatigue) are used to support a more specific finding regarding diabetes disease. For instance, the "Range of Symptoms" shown in Figure 4 illustrate the range for each symptom utilised in the fuzzification process. The green line in Figure 5 indicates that the blood sugar level is normal or low, the yellow line represents pre-diabetes or moderate, and the red line represents diabetes or high.

While the graph for fasting glucose is identical, the measurement ranges for random and fasting glucose are distinct. The (1) is an example of a triangle fuzzifier formula that can be used to determine the crisp value.

$$\mu_A(x) = \begin{cases} (1 - \frac{|x_i - x_i'|}{b_i})...(1 - \frac{|x_n - x_n'|}{b_n}) & \text{if } |x_i - x_i'| \leq b_i, i = 1, 2, ..., n \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\{b, i = 1, ..., n\}$ are positive parameters.

```
# Generate fuzzy triangular membership funct
rbg_lo = fuzz.trimf(x_rbg, [0, 4, 8]) # rand
rbg_md = fuzz.trimf(x_rbg, [7.8, 9.4, 11])
rbg_hi = fuzz.trimf(x_rbg, [10.8, 14.9, 19])
fbg_lo = fuzz.trimf(x_fbg, [0, 3.1, 6.2]) #
fbg_md = fuzz.trimf(x_fbg, [6, 6.5, 7])
fbg_hi = fuzz.trimf(x_fbg, [6.8, 9.9, 13])
wl_no = fuzz.trimf(x_wl, [0, 0.6, 1.2]) # we
wl_yes = fuzz.trimf(x_wl, [0.8, 1.4, 2])
oh_no = fuzz.trimf(x_oh, [0, 0.6, 1.2]) # of
oh_yes = fuzz.trimf(x_oh, [0.8, 1.4, 2])
dz_no = fuzz.trimf(x_dz, [0, 0.6, 1.2]) # of
dz_yes = fuzz.trimf(x_dz, [0.8, 1.4, 2])
tr_no = fuzz.trimf(x_tr, [0, 0.6, 1.2]) # of
tr_yes = fuzz.trimf(x_tr, [0.8, 1.4, 2])
bv_no = fuzz.trimf(x_bv, [0, 0.6, 1.2]) # bl
bv_yes = fuzz.trimf(x_bv, [0.8, 1.4, 2])
res_lo = fuzz.trimf(x_res, [0, 0.6, 1.2]) #
res_md = fuzz.trimf(x_res, [1, 1.5, 2])
res_hi = fuzz.trimf(x_res, [1.8, 2.4, 3])
```

Figure 4. Python code for triangle membership function generation

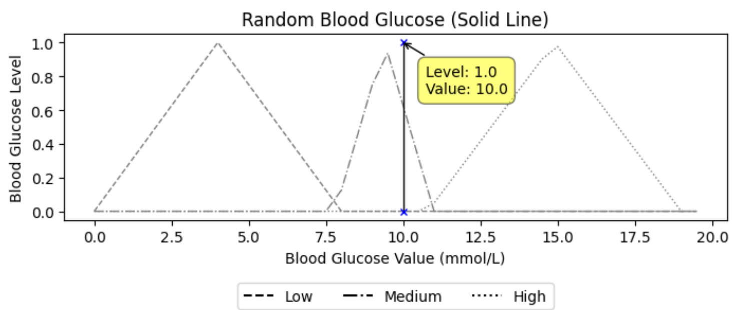

Figure 5. Membership function – random blood glucose

4.2. Evaluation of fuzzy inference

The inference engine applies the inference rules to the fuzzy input to produce the fuzzy output [25]. Specifically, the inference rules are employed to evaluate the linguistic values and map them to a fuzzy set, which then requires the defuzzification procedure to be converted into a crisp value. The “OR” operator is utilised in this study to determine the disjunction of rule antecedents, as seen in Table 1. The definition of the fuzzy “OR” operator is given as (2).

$$\mu_A \cup_B = max[\mu_A(x), \mu_B(x)] \quad (2)$$

The membership in class A is specified as $\mu_A(x)$ while $\mu_A(x)$ represents the membership in class B. The rules depict the rules that were employed in this study, which totals ten rules with seven attributes as shown in Table 1.

Table 1. Sample of rules obtained from experts

Rules Number	RGB	FBG	Weight Loss	Often hunger	Dizziness	Fatigue	Blurry Vission
R1	Low	Low	No	No	No	No	No
R2	Medium	Low	No	No	No	No	No
R3	Medium	Medium	No	No	No	No	No
R4	High	Medium	No	No	No	No	No
R5	High	High	No	No	No	No	No
R6	Low	Low	Yes	Yes	Yes	Yes	Yes
R7	Medium	Low	Yes	Yes	Yes	Yes	Yes
R8	Medium	Medium	Yes	Yes	Yes	Yes	Yes
R9	High	Medium	Yes	Yes	Yes	Yes	Yes
R10	High	High	Yes	Yes	Yes	Yes	Yes

4.3. Defuzzification

Aggregation is the process of combining the output of all fuzzy rules into a single fuzzy set. In contrast, defuzzification is the converse of fuzzification, in which the aggregated output of a fuzzy logic system is turned to its crisp output. This study used the centroid to define a vertical line and divide a compiled set into two equal parts. The equation for calculating the centre of gravity (COG) is given in (3). Defuzzification method (COG) finds a point representing the centre of gravity of the fuzzy set $\mu_A(x)$, on the interval, and a and b is a reasonable estimate can be obtained by calculating it over a sample of points. As depicted in Figure 6, we translate (3) into Python code to implement defuzzification in this study.

$$COG = \frac{\int_b^a \mu_A(x)x \quad dx}{\int_b^a \mu_A(x)dx} \tag{3}$$

```
# Aggregate(combine) all three output membership functions together
# Aggregate all output - max
aggregated = np.fmax(reslo, np.fmax(resmd, reshi))

# Calculate defuzzified result / Defuzzify using centroid
res = fuzz.defuzz(x_res, aggregated, 'centroid')
res_activation = fuzz.interp_membership(x_res, aggregated, res)  # i
```

Figure 6. Python code to implement the procedure of defuzzification

5. INITIAL DEMONSTRATION

This section will describe diabetes diagnosis as an example of the proposed framework for developing complicated medical apps in Python. This experiment verifies the accuracy of the system's outputs based on data obtained from expert and user inputs. Figure 7 demonstrates that the user must enter the system's required information first. The input field indicates where the user must enter the necessary information. After inputting the data, the system will compute and generate the result based on user input using the approach described in Figure 8. The output displays the user-entered information and the resulting text. Also, the resultant graph depicts (as can be seen in Figure 9) the outcome and value for both random and fasting blood glucose levels.

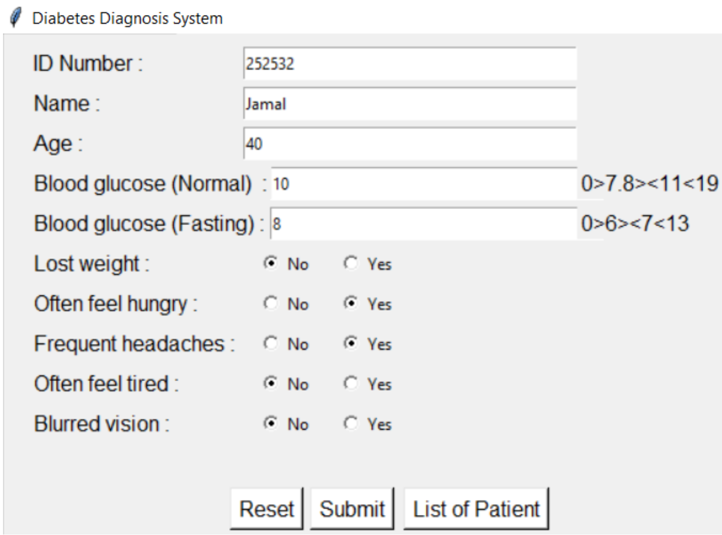

Figure 7. Input Field

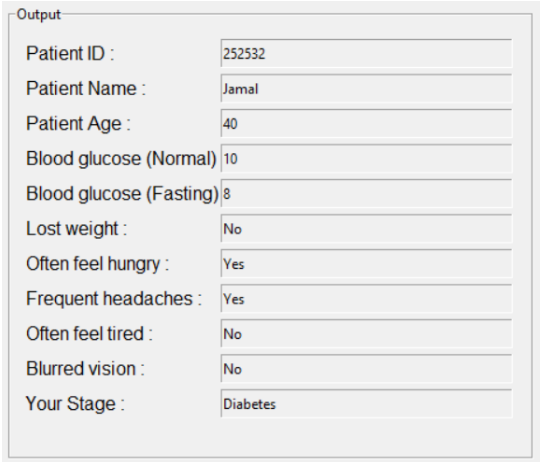

Figure 8. Output interface

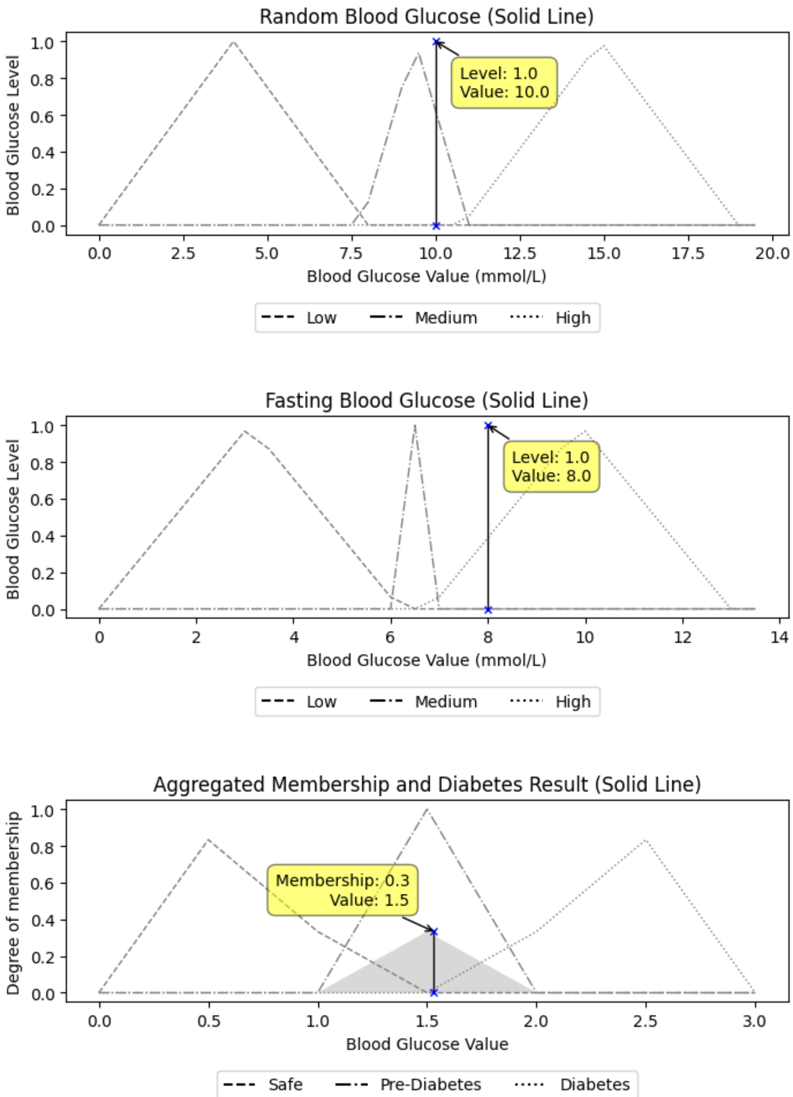

Figure 9. Membership functions: result

6. RESULTS AND DISCUSSION

As described in section 5, we have demonstrated the application of the proposed framework for diabetes diagnosis via the Python program. Figure 9 indicates that the user has been diagnosed with diabetes. The user entered the values 10 for random blood glucose and 9 for fasting blood glucose after comparing the values of random and fasting blood glucose with the graph, as in Figure 9. In contrast, 10 is regarded as medium or pre-diabetes for unanticipated blood glucose situations, but a value of 9 is deemed high or diabetes for fasting blood glucose.

Also, the result is shown in two methods, first by using the triangular graph as in Figure 9. This graph is separated into three triangle shapes with its colour. The left side of the triangle, green, is considered safe from diabetes. In contrast, the middle side of the triangle is considered pre-diabetes, and the colour given is yellow as a warning. Lastly, it is the right side of the triangle, which is diabetes; this triangle is set with red colour as it is considered dangerous, and the user who is diagnosed with this should see the doctor. Another method is translating those values from the trapezoid graph into easy-to-read text in Figure 8. Easy to Read Text is separated into three status: safe, pre-diabetes, and diabetes. Hence, the user's inputs determine this status. In order to validate rules based on fuzzy expert systems, eleven experts were utilised. The expert-derived principles shown in Table 1 are employed by this system to forecast diabetes status. Based on current evidence, medical experts concur on the standards for the diagnosis of diabetes, indicating that the system's calculations and conclusions may be correct.

7. CONCLUSION

In conclusion, this study provided a self-diagnosis of the fuzzy expert system for diabetic disease utilising Python programming. Based on current research evidence, the diabetes diagnosis prototype has shown promising results since it can determine a person's diabetes status based on seven attributes. This aims to examine the relationship between the output from the input variables using a set of fuzzy rules. Also, in this study, seven attributes are merged to form a single set of ten fuzzy rules. These rules will be applied by their rationale. These rules have distinct inference behaviours, each corresponding to various intended applications. In addition, this study was created using Python, and Tkinter will provide insight for the researcher to model the fuzzy logic system for real-world application. The inclusion of an interface simplifies system administration for the user. In this study, a medical professional or any system user can save their or other user data by providing the needed information and saving it for future reference. While this study gives intriguing insights on modelling the fuzzy expert system for real-world applications using the Python programming language, additional research is required, such as using more complex applications with larger datasets and implementing different defuzzification strategies. This may provide a more comprehensive understanding of fuzzy expert system modelling, particularly for programmers utilising languages other than Python.

ACKNOWLEDGEMENTS

This work was supported by a Fundamental Research Grant Scheme (FRGS) from the Ministry of Higher Education in Malaysia (Reference No.: FRGS/1/2021/ICT02/UITM/02/1).

REFERENCES

[1] M. Sharifi-Rad *et al*., "Lifestyle, oxidative stress, and antioxidants: Back and forth in the pathophysiology of chronic diseases," *Frontiers in physiology*, vol. 11, p. 694, 2020, doi: 10.3389/fphys.2020.00694.

[2] H. R. Murphy *et al*., "Characteristics and outcomes of pregnant women with type 1 or type 2 diabetes: a 5-year national population-based cohort study," *The lancet Diabetes & endocrinology*, vol. 9, no. 3, pp. 153–164, 2021, doi: 10.1016/S2213-8587(20)30406-X.

[3] K. Ganasegeran *et al*., "A systematic review of the economic burden of type 2 diabetes in Malaysia," *International journal of environmental research and public health*, vol. 17, no. 16, p. 5723, 2020, doi: 10.3390/ijerph17165723.

[4] A. Budreviciute *et al*., "Management and prevention strategies for non-communicable diseases (NCDs) and their risk factors," *Frontiers in public health*, p. 788, 2020, doi: 10.3389/fpubh.2020.574111.

[5] T. R. B. Razak, R. A. Wahab, and M. H. Ramli, "Dengue notification system using fuzzy logic," in *2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 2013, pp. 231–235, doi: 10.1109/IC3INA.2013.6819179.

[6] A. J. Dahalan, T. R. Razak, M. H. Ismail, S. S. M. Fauzi, and R. A. J. M. Gining, "Heart rate events classification via explainable fuzzy logic systems," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 4, pp. 1036–1047, 2021, doi: 10.11591/ijai.v10.i4.pp1036-1047.

[7] T. R. Razak, N. H. Anuar, J. M. Garibaldi, and C. Wagner, "Modelling Hierarchical Fuzzy Systems for Mango Grading via FuzzyR Toolkit," in *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2022, pp. 1–8, doi: 10.1109/FUZZ-IEEE55066.2022.9882553.

[8] W. Helen and E. Bhavya, "A prospective study on effectiveness of sitagliptin on glycemic control, blood pressure and serum lipid profile levels in type-II diabetes mellitus patients," *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 5, pp. 3585–3592, 2021.

[9] R. D. Leslie *et al.*, "Adult-onset type 1 diabetes: current understanding and challenges," *Diabetes Care*, vol. 44, no. 11, pp. 2449–2456, 2021, doi: 10.2337/dc21-0770.

[10] S. M. Jwad and H. Y. AL-Fatlawi, "Types of Diabetes and their Effect on the Immune System," *Journal of Advances in Pharmacy Practices*, vol. 4, no. 1, pp. 21–30, 2022.

[11] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965, doi: 10.1016/S0019-9958(65)90241-X.

[12] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ, USA: Prentice Hall, 2001.

[13] L. Sellami and B. Alaya, "UPSO-FSVRNET: Fuzzy Identification Approach in a VANET Environment Based on Fuzzy Support Vector Regression and Unified Particle Swarm Optimization," *International Journal of Fuzzy Systems*, vol. 25, pp. 743–762, 2023, doi: 10.1007/s40815-022-01408-7.

[14] A. Garrido, "A Brief History of Fuzzy Logic," *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, vol. 3, no. 1, pp. 71–77, 2012.

[15] T. R. Razak, J. Garibaldi, C. Wagner, A. Pourabdollah, and D. Soria, "Towards a Framework for Capturing Interpretability of Hierarchical Fuzzy Systems - A Participatory Design Approach," in *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 5, pp. 1160–1172, 2021, doi: 10.1109/TFUZZ.2020.2969901.

[16] Y. Peng, Y. Zhang, and M. Hu, "An empirical study for common language features used in python projects," in *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2021, pp. 24–35, doi: doi: 10.1109/SANER50967.2021.00012.

[17] E. Zolduoarrati and S. A. Licorish, "On the value of encouraging gender tolerance and inclusiveness in software engineering communities," *Information and Software Technology*, vol. 139, p. 106667, 2021, doi: 10.1016/j.infsof.2021.106667.

[18] D. A. Grimes and K. F. Schulz, "Refining clinical diagnosis with likelihood ratios," *The Lancet*, vol. 365, no. 9469, pp. 1500–1505, 2005, doi: 10.1016/S0140-6736(05)66422-7.

[19] S. Mishra, H. K. Tripathy, H. Kumar Thakkar, D. Garg, K. Kotecha, and S. Pandya, "An explainable intelligence driven query prioritization using balanced decision tree approach for multi-level psychological disorders assessment," *Frontiers in Public Health*, vol. 9, p. 795007, 2021, doi: 10.3389/fpubh.2021.795007.

[20] N. Yang *et al.*, "Intelligent Data-Driven Decision-Making Method for Dynamic Multisequence: An E-Seq2Seq-Based SCUC Expert System," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3126–3137, 2021, doi: 10.1109/TII.2021.3107406.

[21] K. R. Srinath, "Python - the fastest growing programming language," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 12, pp. 354–357, 2017.

[22] S. Masis, *Interpretable Machine Learning with Python: Learn to build interpretable high-performance models with hands-on real-world examples*. Birmingham, United Kingdom: Packt Publishing Ltd, 2021.

[23] Q. Charatan and A. Kans, "Python Graphics with Tkinter," in *Programming in Two Semesters*, Cham, Switzerland: Springer, 2022, pp. 211–254, doi: 10.1007/978-3-031-01326-3_11.

[24] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[25] S. Rizvi, J. Mitchell, A. Razaque, M. R. Rizvi, and I. Williams, "A fuzzy inference system (FIS) to evaluate the security readiness of cloud service providers," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–17, 2020, doi: 10.1186/s13677-020-00192-9.

BIOGRAPHIES OF AUTHORS

Tajul Rosli Razak is currently working at the School of Computing Sciences, College of Computing, Informatics and Media Universiti Teknologi MARA, Malaysia. He received a Bachelor of Information Technology (Hons) specialising in artificial intelligent from the Universiti Utara Malaysia and a Master of Science (Intelligent Systems) from the Universiti Utara Malaysia. He then obtained his Ph.D. in Computer Science from the University of Nottingham, United Kingdom. While pursuing his Ph.D., He was a member of Intelligent Modelling & Analysis (IMA) and Laboratory for Uncertainty in Data and Decision Making (LUCID) research groups. He is currently a member of the IEEE-CIS Task Force on Explainable Fuzzy Systems. His research interests centre around intelligence systems, fuzzy logic systems and hierarchical fuzzy systems, decision support systems and interpretability. He can be contacted at email: tajulrosli@uitm.edu.my.

Ahmad Zia Ul-Saufie is an associate professor of statistics and the head of the School of Mathematical Sciences, Universiti Teknologi MARA. He has extensive research interests covering various interdisciplinary perspectives concerning the theory and practice of Data Science and Statistics. He has published well over 100 peer-reviewed scientific international and national publications in multidisciplinary research areas, including applied statistics, machine learning, neural networks, environmental analytics, survey study, sport science analytics, medical knowledge analysis, sensors and IoT, and system development. But his current research passion is data sciences for environmental and sports sciences. He can be contacted at email: ahmadzia101@uitm.edu.my.

Mohamad Hanis Yusoff was a research student from the Faculty of Computer and Mathematical Sciences and is under guidance and direction from the researchers of Applied Computing and Technology Research Group. He obtained his first degree in information technology from UiTM in 2021. He won a silver medal in the International Malaysia-Indonesia-Thailand Symposium on Innovation and Creativity (iMITSIC 2021) with the title project "Python-Based: Self-Diagnosis of Diabetes Disease Using Fuzzy Expert Systems". He can be contacted at email: mohamadhanis.yusoff@gmail.com.

Mohammad Hafiz Ismail is a senior lecturer and researcher from the Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perlis Branch, Malaysia. He obtained his first degree in data communication and networks and his master's in information technology. His primary research interest is mobile and pervasive computing, and he is actively involved in mobile technology solutions in his community. He can be contacted at email: mohammadhafiz@uitm.edu.my.

Shukor Sanim Mohd Fauzi is a Rector at UiTM Perlis Branch, and a faculty member of the Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perlis Branch, Malaysia. He received a Master of Science (Computer Science - Real-Time Software Engineering) from the Centre for Advanced Software Engineering, Universiti Teknologi Malaysia. He then obtained his Ph.D. in Software Engineering from the University of New South Wales (UNSW). He can be contacted at email: shukorsanim@uitm.edu.my.

Nurul Ain Mohd Zaki a senior lecturer and researcher from Studies for Surveying Science & Geomatics, College of Built Environment (CBE), Universiti Teknologi MARA, Perlis Branch. She obtained her Ph.D. at Universiti Teknologi MARA, Shah Alam. She is also an associate fellow researcher for the Institute of Biodiversity and Sustainable Development (IBSD) and a member of the International Society for Photogrammetry and Remote Sensing. Her primary research interest is geographical information science, data modelling, light detection and ranging (LIDAR), aboveground biomass, carbon stocks, and remote sensing for forestry. She can be contacted at email: nurulain86@uitm.edu.my.