Vol. XVI & Issue No. 07 July - 2023
INDUSTRIAL ENGINEERING JOURNAL

FUZZY LOGIC BASED APPROACH TO PREDICT SOFTWARE MAINTAINABILITY

Anita Devi
Saurabh Charaya
Mukesh Maan

Abstract

Software maintainability is proven to be one of the important factors to predict software Quality. Nowadays users are more focused on quality rather than quantity. In order to predict software maintainability at an early stage with better accuracy, soft computing techniques namely fuzzy logic is widely used and helpful in minimizing cost and maintenance time. In this paper, we proposed and implemented a fuzzy model using four parameters based on the ISO9126 Quality model viz: analyzability, changeability, stability, and testability to predict software maintainability level. Each input parameter is divided into Low, Moderate, and High whereas the output parameter is divided into Very_Low, Low, Moderate, High, and Very_High. Proposed model yields software maintainability level with better accuracy.

Keywords: Software Maintainability, Fuzzy Logic, Neural Network, Software Quality

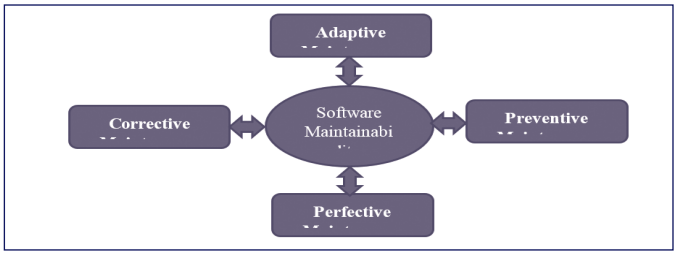
1. INTRODUCTION

In the present scenario, maintainability is one of the key challenges for software developers. Maintenance is basically required to ensure that the software continues to satisfy user requirements. Maintainability may be defined as a "qualitative indication of the ease with which existing software can be corrected, adapted, or enhanced" [15, 16]. As per the ISO9126 Quality Model [2], Maintainability may be defined as "A set of attributes that bear on the effort needed to make specified modifications (which may include corrections, improvements, or adaptions of software to environmental changes and changes in the requirement and functional specifications)". Software Maintenance may also be defined as "the modification of a software product after delivery to correct faults, to expand performance or other qualities" [4, 8, 13].

1.1 Types of Software Maintenance

There are basically four types of Software maintainability, which may be categorized as adaptive maintenance, preventive maintenance, perfective maintenance, and corrective maintenance as shown in Figure 1. K. K. Aggarwal et. al [7] developed a fuzzy model using 04 input parameters to measure software maintainability. The Developed model is validated and results are found to be satisfactory.

Fig 1: Types of Software Maintenance



Yogesh Singh et al. [18] presented a fuzzy model using 04 major factors namely Readability of Source Code, UoS, Documentation Quality, and ACC to forecast maintainability, and results are found to be authenticated and satisfactory. Ritu and Om Prakash Sangwan [14] discussed various software quality estimation techniques using soft computing techniques (FL, NN, and ANFIS). It is also observed that software maintainability played a vital role in estimating software quality and found that Neural Network and Fuzzy Logic techniques were widely used for software quality estimation. For the estimation of software maintainability, there is a need to find out a group of characteristics/factors/attributes that are needed. Lots of factors affect the maintenance part of software development and need to be specially taken care of with utmost priority to safeguard effective maintenance. In this paper, a fuzzy model has been proposed to forecast software maintainability levels using four input parameters analyzability, changeability, stability, and testability. The organization of this paper is as follows. Related work is presented in section 2. In section 3 of the paper, we have discussed the various factors that affect software maintainability to a greater extent. The proposed methodology has been discussed in section 4 of the paper. A software maintainability model has been designed in section 5 of the paper. Implementation of the proposed model was discussed in section 6 of the paper. Section 7 of the paper is presented with the conclusion and future work.

2. RELATED WORK

We have studied several research papers including research papers from IEEE, Springer, and other reputed Journals and Conferences to evaluate software maintainability using soft computing techniques. A brief discussion of a few of the papers is presented here in this section. Grover et al. [6] proposed a fuzzy model for a component-based system. They have

used Understandability, Testability, Reusability, Interface Complexity, and Trackability as input parameters to predict software maintainability. The proposed model was then applied to a classroom-based project and the results are found to be satisfactory. Neha Goel et al. [11] developed a fuzzy model to estimate the maintainability of an object-oriented system. Results obtained proved that the proposed model suitable for the prediction of software maintainability for Object-oriented systems. Amrendra Pratap et al. [1] used fuzzy-logic techniques to forecast software maintainability. To predict software maintainability four attributes namely Adaptability, Document Quality, Complexity, Readability, and Understandability were used as input, and experimental results proved the successful implementation of the proposed model. Mamta and Amandeep Kaur [10] applied fuzzy logic and Artificial Neural Networks to estimate software maintainability for component-based systems. They have identified five major input parameters namely documentation, modifiability, integrability, testability, and coupling between components to estimate software maintainability. Based on expert judgment, 243 rules were fired with fuzzy logic. The triangular membership function was used for fuzzification and the Centre of Gravity (CoG) was used for defuzzification. Based on the output received from the fuzzy model, a data set of 150 inputs along with output was prepared and ANN is used to predict the maintainability level. The proposed ANN model was trained well and yielded satisfactory results with MARE 24% and MRE 0.009%. Deeksha and Aman Kaushik [3] focused on the measurement and prediction of quality attributes of maintainability and worked on the systematic review of effective maintainability prediction to improve the overall software quality. Rachna Jain and Arun Sharma [12] carried out a detailed literature review related to the reliability and maintainability of software and observed a variety of soft computing and optimization techniques used by different researchers for evaluating the reliability and maintainability of the software system. Shivani Kundu and Kirti Tyagi [17] presented a comparison and assessment of maintenance efforts using soft computing techniques and found them to be appropriate with reasonable results. Lov and Rath [9] developed a software maintainability prediction model using a hybrid neural network and fuzzy logic approach using 10 different object-oriented static source code metrics as input. The proposed method was applied to UIMS and QUES data sets and the results indicate that the neuro-fuzzy approach yielded a better result. Gokul Yenduria [5] discussed about the snags and potential aids for estimating software maintainability using soft computing techniques. Based on the literature, it is concluded that researchers have used a number of soft computing techniques with different input attributes to predict software maintainability. However, it is not clear about the minimum no of attributes required to estimate software maintainability and which soft computing technique is better than the other one. Here, the fuzzy logic approach has been used to evaluate software maintainability using four input parameters taken from the ISO9126 quality model.

3. FACTORS AFFECTING SOFTWARE MAINTAINABILITY

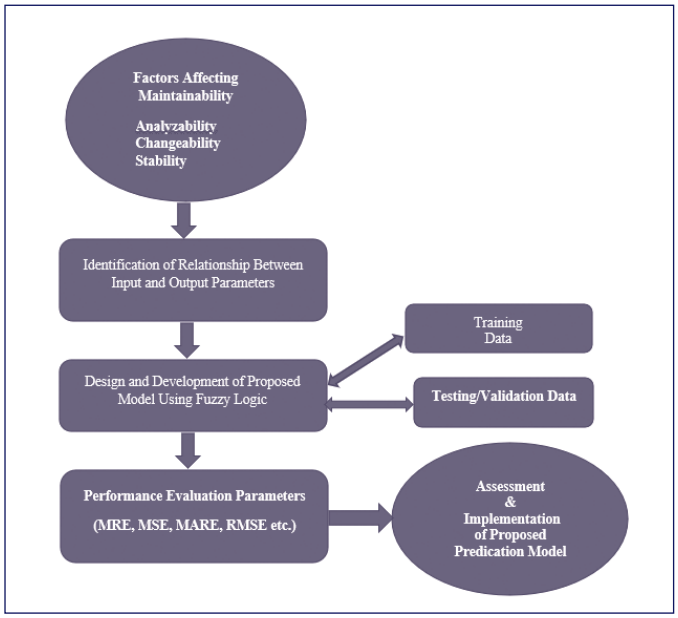
Based on the review of the related work discussed in section 2 of the paper, it is concluded that there is a need to identify the factors that can affect software maintainability to a larger extent and are consistent among various models. Based on the literature survey on software quality models, and attributes considered, we have identified the four major quality sub-characteristics that affect quality attributes namely Analyzability, Changeability, Stability, and Testability [2]:

In order to estimate the maintainability of software, there is a need to establish a relationship between input and output parameters to achieve a predefined goal. In this paper, with the help of fuzzy logic, we have proposed to design a system that forecasts software maintainability depending on the relationship between input and output. The kind of relationship between input and output is determined with the help of a domain expert. For designing and experimental purposes, we have used the Fuzzy Logic Designer of MATLAB [19].

4. PROPOSED METHODOLOGY

There is no fixed criteria or formula to predict or evaluate maintainability. In this paper, a generic software Maintainability model has been proposed as shown in Figure 2.

Fig 2: Generic Software Maintainability Model



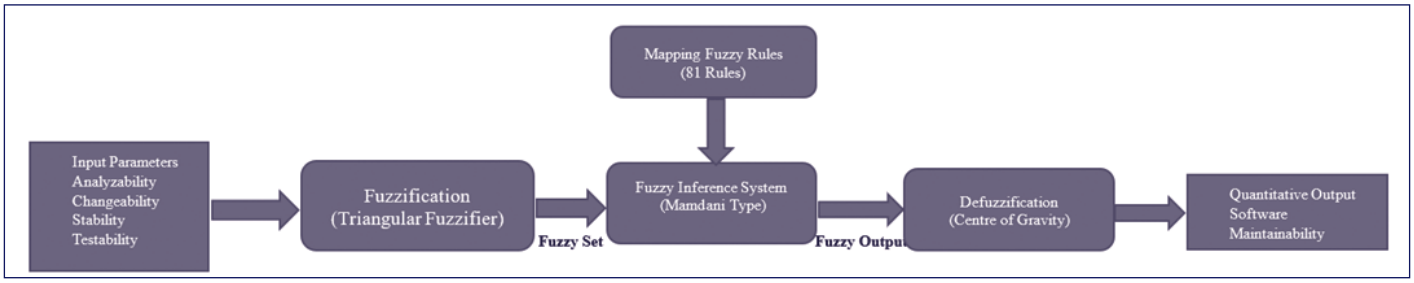
Firstly, there is a need to identify factors that affects software maintainability. In this paper, we have identified four factors based on the ISO 9126 quality model as shown in Figure 2 to predict software maintainability level. Each input parameter is divided into Low, Moderate, and High whereas the output parameter has been categorized as Very_Low, Low, Moderate, High, and Very_High. Secondly, to find out the relationship between the input and output parameters, 81 rules have been drawn with the help of domain experts. Further, a fuzzy model has been designed and implemented with 81 rules and the output is also validated mathematically.

5. PROPOSED SOFTWARE MAINTAINABILITY FUZZY MODEL

There are lots of methodologies that exist to predict software

maintainability, but kind the methodology used to evaluate software maintainability varies from data set to data set. Here in this paper, a fuzzy logic-based approach to predict software maintainability levels has been proposed using four major parameters based on the ISO9126 quality model. Since the input parameters are linguistics in nature, therefore fuzzy model works well under this situation. Moreover, fuzzy logic can be used where we have no data or little data exists. Fuzzy logic works with explicit knowledge. For designing and the experimental implementation, we used the fuzzy tool of MATLAB. We have considered four major attributes namely analyzability, changeability, stability, and testability as inputs to the proposed fuzzy model to predict software maintainability. Each input parameter is categorized into Low, Moderate, and High whereas the output parameter maintainability is also categorized into Very Low, Low, Moderate, High, and Very high. The value of the input and output parameters lies in the interval [0,1]. It is assumed that Very Low maintenance is good, whereas Very High maintenance is poor. The proposed fuzzy model with four inputs and one output is shown in Figure 3.

Fig 3: Proposed Software Maintainability Fuzzy Model



A Mamdani fuzzy inference system has been used to map the input and the output. The triangular membership function has been used for the fuzzification of input and output variables. Based on the feedback received from domain experts, 81 rules have been manually prepared as shown in Table 1. The Center of Gravity (COG), has been used for defuzzification.

Table 1: Proposed 81 Fuzzy Rules

S. No.	Analyzability	Changeability	Stability	Testability	Maintainability
1	L	L	L	L	VH
2	L	L	L	M	VH
3	L	L	M	L	H
4	L	L	M	M	H
5	L	L	L	H	H
. .					
50	M	H	L	H	L
51	M	H	H	L	L
52	M	H	H	H	VL
53	M	H	H	M	L
54	M	H	M	H	L
. .					
77	H	M	L	H	L
78	H	M	H	L	L
79	H	M	L	L	M
80	H	M	H	M	L
81	H	M	M	H	L
L = LOW, VL = VERY LOW, M= MODERATE, H= HIGH, VH = VERY HIGH					

IMPLEMENTATION OF THE PROPOSED MODEL

Maintainability is one of the crucial parts of every software development process. So, we have considered four important key parameters as discussed to predict maintainability level. Fuzzy Logic Designer Mamdani type1 has been used to evaluate the maintainability. The four parameters are passed as input to the system and the maintainability level is taken as output.

5.1 Fuzzification: The triangular Membership function has been used for the fuzzification of all inputs and output

parameters as shown in Figure 4 and Figure 5 respectively. Each input parameter is categorized into Low, Moderate, and High. Whereas the output parameter maintainability is also categorized into Very Low, Low, Moderate, High, and Very high. The value of the input and output parameters lies in the interval [0,1].

Fig 4: Membership Function for Input Parameters

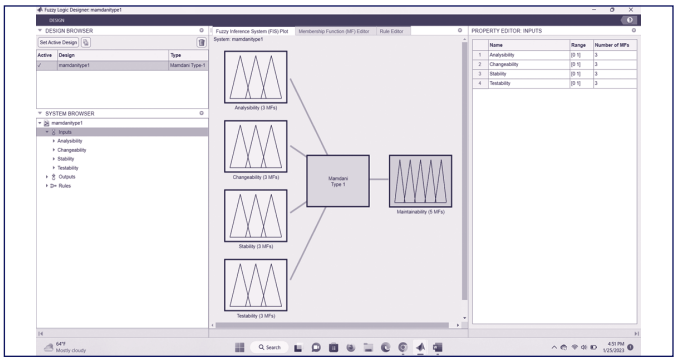
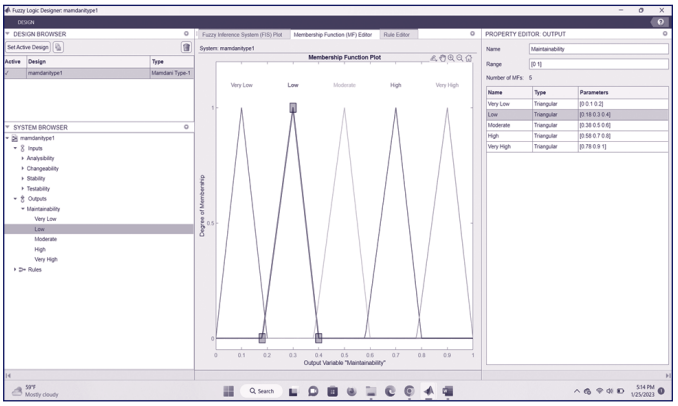
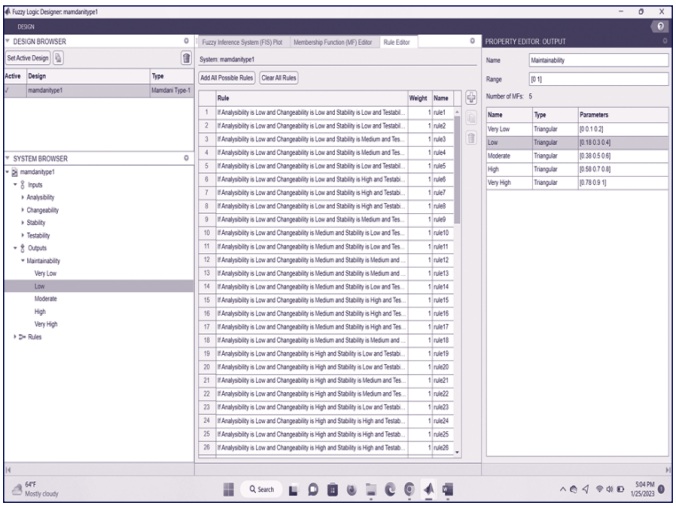


Fig 5: Membership Function for Output Parameter



6.2 Fuzzy Rules: As there are four Input parameters (Analyzability, Changeability, Stability, and Testability) and each input parameter is divided into Low, Moderate, and High. So, there will be 3^4 i.e., 81 Rules. Based on domain expertise, 81 rules have been finalized and added to the Rule Editor and after that, we can fire each and every rule to check the effect of these rules on maintainability. In Figure 6, we can see all the added rules in the Rule Viewer.

Fig 6: Fuzzy Rule Viewer



With the help of the Ruler Inference System, we can Check how the value of maintainability is affected by the values of the input parameters. Rules are fired and each corresponding value can be seen in the Ruler Inference System. Five rules are fired randomly here to check the maintainability of the system by considering the different values of input parameters, as shown below: -

Rules	Input Parameters				Output Parameter
	Analyzibility	Changeability	Stability	Testability	Maintainability
Rule 1	0.12(L)	0.25(L)	0.2(L)	0.1(L)	0.892 (V. High)
Rule 70	0.8(H)	0.12(L)	0.09(L)	0.16(L)	0.692 (High)
Rule 28	0.56(M)	0.55(M)	0.45(M)	0.57(M)	0.493 (Moderate)
Rule 62	0.8(H)	0.85(H)	0.75(H)	0.52(M)	0.293 (Low)
Rule 55	0.78(H)	0.8(H)	0.85(H)	0.79(H)	0.1 (V. Low)

All above randomly fired rule values can be verified by the RIS (Ruler Inference System) as shown in figure 7 to figure 11 respectively.

Fig. 7: Rule 1 with Maintainability Level (Very High)

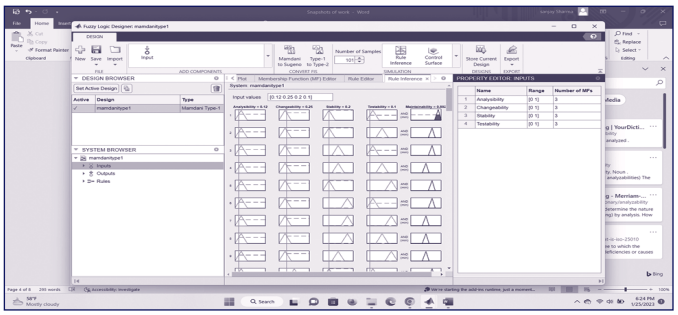


Fig. 8: Rule 70 with Maintainability Level (High)

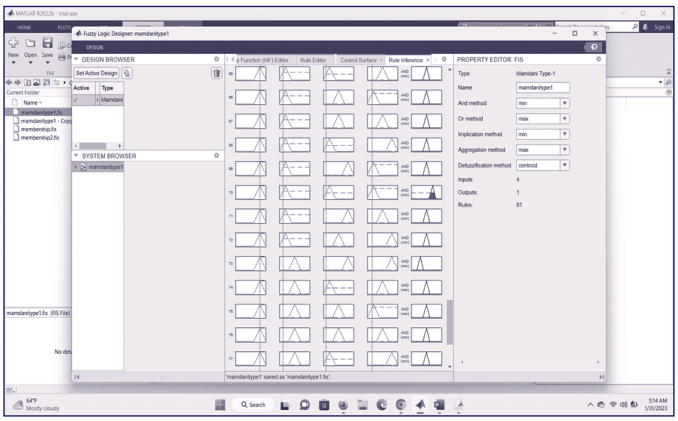


Fig. 9: Rule 28 with Maintainability Level (Moderate)

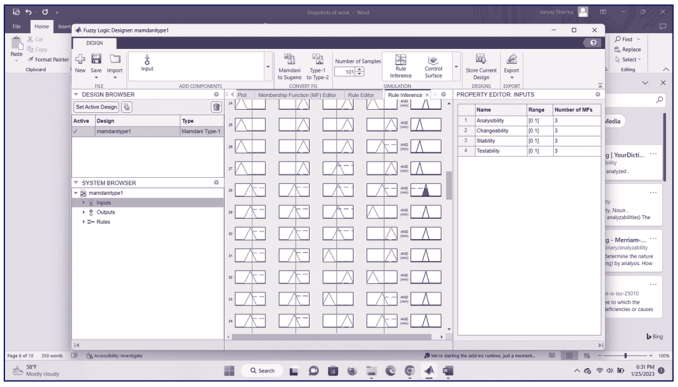


Fig. 10: Rule 62 with Maintainability Level (Low)

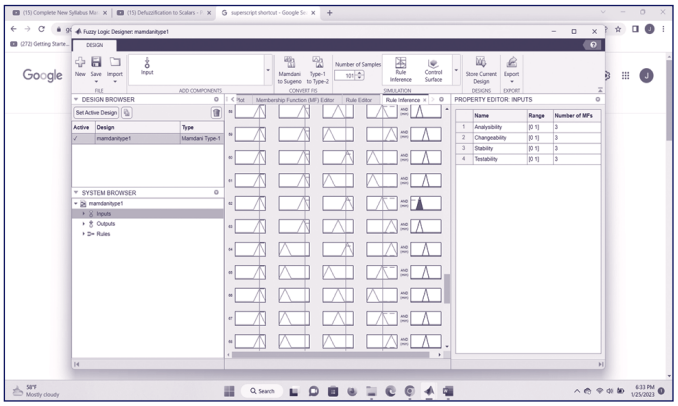
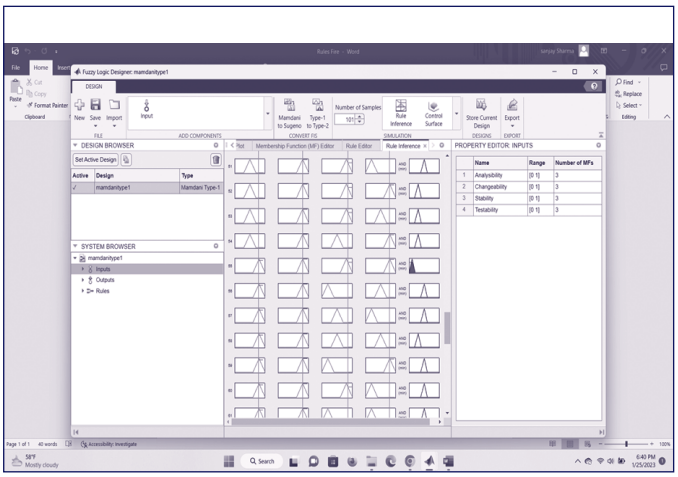


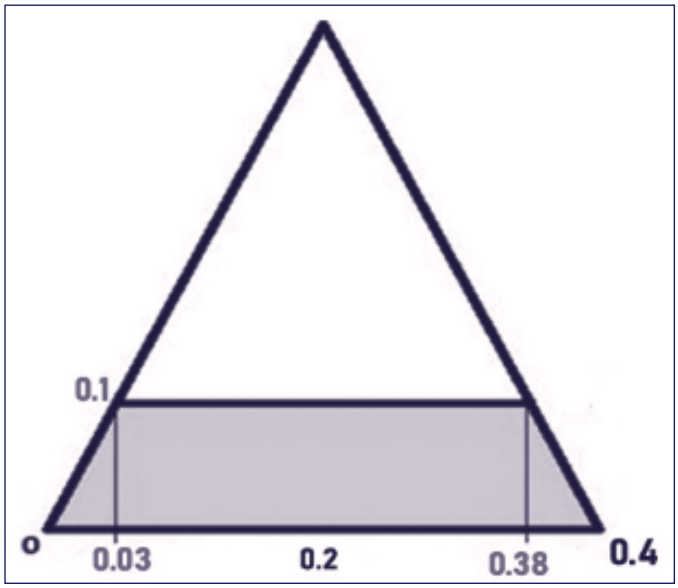
Fig. 11: Rule 62 with Maintainability Level (Very Low)



Therefore, it is very clear from the above figures that each and every rule among a set of 81 Rules is verified by the Ruler Inference System. The proposed model helps the user and the software developer to predict in advance the maintainability level of software by considering these input parameters (Analyzability, Changeability, Stability, and Testability).

6.3 Defuzzification: The Centroid method is used here for Defuzzification. 'Min' implication is employed for estimating rule output. Minimum of "Analyzability= 0.12, Changeability = 0.25, stability = 0.2, Testability = 0.1 is 0.1. Thus, the fuzzified output of this rule can be shown in figure 12:

Fig. 12: Fuzzified Output



$$\text{MAINTAINABILITY} = \frac{\int_0^{0.4} x\mu_x(x)dx}{\int_0^{0.4} \mu_x(x)dx}$$

$$\text{MAINTAINABILITY} = \frac{\int_0^{0.03} x.(10/3\,x)dx + \int_{0.03}^{0.38} x.(0.1\ )dx + \int_{0.38}^{0.4} x.(-\ 5x+2)dx}{\int_0^{0.03} (10/3)\,xdx + \int_{0.03}^{0.38} 0.1\ dx + \int_{0.38}^{0.4} (-\ 5x+2)dx}$$

$$\text{MAINTAINABILITY} = \frac{\int_0^{0.03} (10/3\,x^2)dx + \int_{0.03}^{0.38} x.(0.1\ )dx + \int_{0.38}^{0.4} (-\ 5x^2+2x)dx}{\int_0^{0.03} (10/3)\,xdx + \int_{0.03}^{0.38} 0.1\ dx + \int_{0.38}^{0.4} (-\ 5x+2)dx}$$

$$\text{MAINTAINABILITY} = \frac{2.137225}{3} * \frac{1}{0.8625}$$

$$\text{MAINTAINABILITY} = 0.8$$

It is concluded that the overall measures indicate that all considered input factors have strong relation with maintainability.

6. CONCLUSION AND FUTURE WORK

In this paper, a software maintainability estimation model has been proposed using fuzzy logic. Efforts have been made to identify the major inputs that affect the software's maintainability. The proposed model was trained with input parameters Analyzability, Changeability, stability, and Testability. Triangular membership has been used for the fuzzification of input and output parameters. Based on the expert judgment 81 rules have been finalized and the Mamdani fuzzy inference system has been used to map input and output parameters. In order to validate the proposed approach, we have manually calculated the maintainability value with the help of Center of Gravity (CoG) and found it to be very close to the value as predicted by the fuzzy model. With the help of the fuzzy logic approach, we are able to know, how much an input attribute impacts the maintainability level of software up to what extent. Experimental results obtained clearly indicates that input parameters considered in this paper have a strong relationship with maintainability. In the future, we may explore the possibilities of other soft computing techniques to forecast software maintainability.

REFERENCES

[1]. *A. Pratap, R. Chaudhary, and K. Yadav (2014), "Estimation of Software Maintainability Using Fuzzy Logic Technique," in International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT). IEEE, 2014, pp.486–492.*

[2]. *B. Kitchenhsm and S. L. Pfleeger (1996), "Software Quality: The Elusive Target", IEEE Software, January 1996, pp. 12-21.*

[3]. *Deeksha and Aman Kaushik (2016), "A Systematic Review of Software Maintainability Prediction," RIEEC-2106, Volume 3, Spl. Issue 1 (2016), pp.: 217-219.*

[4]. *Don Coleman and Dan Ash, Bruce Lowther, Paul Oman (2015), "Using Metrics to Evaluate Software System Maintainability," Hewlett-Packard, Micron Semiconductor, University of Idaho, IEEE Computer, vol.27, no.8, pp.44-49, Aug. 1994. Computer Science (IJITCS), vol.7, no.7, pp. 90-10.*

[5]. *Gokul Yenduria, Thippa Reddy and Gadekallu (2022), "A Systematic Literature Review of Soft Computing Techniques for Software Maintainability Prediction: State-of-the-Art, Challenges and Future Directions", arXiv:2209.10131v1 [cs.SE].*

[6]. *Grover, P. S., Kumar, R., and Sharma A. (2007), "Few Useful Considerations for Software Components and Component-Based Systems," ACM SIGSOFT Software Engineering Notes, Vol. 32, Issue 4, pp: 1-5.*

[7]. *K. K. Aggarwal, Yogesh Singh, Parvind Chandra, and Manimala Puri, (2005), "Measurement of Software Maintainability Using a Fuzzy Model,", April 2005, Journal of Computer Science1(4), 2005.*

[8]. *Kurt D. Welker (2001), "The Software Maintainability Index Revisited", Idaho National Engineering and Environmental Laboratory, Crosstalk, Aug. 2001.*

[9]. *Lov Kumar and Santanu Ku Rath (2017), "Software Maintainability Prediction using Hybrid Neural Network and Fuzzy Logic Approach with Parallel Computing Concept," International Journal of System Assurance Engineering and Management, volume 8, pages 1487–1502.*

[10]. *Mamta Punia and Amandeep Kaur (2014), "Software Maintainability Prediction Using Soft Computing Techniques", International Journal of Innovative and Technology, Vol.1, Issue 9.*

[11]. *Neha Goel, Sanjay Kumar Dubey, and Ajay Rana (2012), "Fuzzy Layered Approach for Maintainability Evaluation of Object-Oriented Software System", International Journal of Scientific & Engineering Research, Volume 3, Issue 6.*

[12]. *Rachna Jain and Arun Sharma (2016) , "Critical Insight into Maintainability and Reliability Assessment Aspects of Software Systems Using Soft Computing Approaches," I J C T A, (2016) 9(20), 2016, pp. 81-88.*

[13]. *Rikard Land (2002), "Measurement of Software Maintainability," Department of Computer science, Malardalen University, ARTES Graduate Student Conference, March 2002.*

[14]. *Ritu and Om Prakash Sangwan (2017), "Software Quality Estimation Using Soft Computing Techniques," International Journal of Innovations & Advancement in Computer Science IJIACS (ISSN 2347 – 8616), Volume 6, Issue 5 May 2017.*

[15]. *Schneidewind, N (1999), "Measuring and Evaluating Maintenance Process Using Reliability, Risk, and Test Metrics," IEEE Transactions Software Engineering, Vol. 25, No. 6, November-December 1999.*

[16]. *Schwaber, K (2002), "Agile Processes and Self-Organization", Agile Alliance.*

[17]. *Shivani Kundu and Kirti Tyagi (2016), "Effort Estimation of Software Maintainability Using Soft Computing Techniques: a Critical Literature Survey," World Applied Sciences Journal 34 (6): 733-742, 2016, ISSN 1818-4952, © IDOSI Publications, 2016, DOI: 10.5829/idosi.wasj.2016.34.6.47.*

[18]. *Yogesh Singh, Pradeep Kumar Bhatia, and Om Prakash Sangwan (2009) "Predicting Software Maintenance using Fuzzy Model," ACM SIGSOFT Software Engineering Notes, Volume 34 Issue 4, pp.: 1-6.*

[19]. *https://www.mathworks.com/discovery/what-is-matlab.html.*

AUTHORS

Anita Devi, Department of Computer Science & Engineering, Om Sterling Global University, Hisar, Haryana, India
Email: anita0863@gmail.com

Saurabh Charaya, Department of Computer Science & Engineering, Om Sterling Global University, Hisar, Haryana, India
Email: saurabh.charaya@gmail.com

Mukesh Maan, Department of Computer Science & Engineering, Indian Institute of Information Technology, Sonipat, Haryana
Email: mukesh.maan@iiitsonepat.ac.in