Turkish Journal of Computer and Mathematics Education Vol.12 No.11 (2021), 807-817

Research Article

Software Quality Prediction Method Using Fuzzy Logic

Ritu[1], O.P. Sangwan[2]

[1]Department of CSE Guru Jambheshwar University of Science and Technology, Hisar
[2]Department of CSE Guru Jambheshwar University of Science and Technology, Hisar
[1]rituchopra1984@ gmail.com, [2]sangwan0863@gmail.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: Software Quality is the key priority of today's marketplace and software development organization to which a system, technique, or factor meets particular requirements and conditions. Soft computing techniques play a vital role in developing software engineering applications. In this paper, we have identified five parameters: Reliability, Efficiency, Usability, Maintainability, and Portability for accessing the level of quality of software. A fuzzy logic-based intelligent identification methodology has been proposed to access the quality of particular software-based on five parameters. The proposed identification scheme takes these five parameters as input and predicts the quality of the software using the fuzzy rule base which is generated using various studies. As this scheme takes five inputs and each input is divided into three regions i.e. 'Low', 'Medium', 'High' and thus a total of 35 i.e. 243 rules has been generated to analyze the software quality. Furthermore, Mamdani fuzzy model has been used as the reference model. To show the effectiveness of the proposed methodology, simulation results have been performed in MATLAB, which shows that the software's quality closely matches with the actual one.

Keywords: Software quality, Fuzzy Logic (FL), Reliability, Efficiency, Usability, Maintainability and Portability.

1. Introduction

Software engineering is to create high-quality software products that can be created on time and within budget. As software has become very important in this digital world, the quality of particular software plays an important role while selecting software among various parameters [1]. Software Quality [2][3] is a key factor that is considered in software development. Software quality depends on many factors which include cost as well. Some definitions of software quality are as follows:

- Software Quality is the degree to which a system, method, or component meets particular requirements and conditions [4].
- According to Dr. Barry Boehm [5], quality is defined as achieving high levels of user satisfaction, portability, maintainability, robustness, and fitness for use.
- Tom McCabe [5], the software complexity experts, defines quality as "High levels of user satisfaction and low defect levels, often associated with low complexity".

Thus, the automatic quality estimation can be very useful [6] while selecting software where customers put their needs in terms of inputs and the quality of the software will be shown to the customers. Then, they can choose the software based on the quality estimated.

Various research have proposed software quality models to predict software quality and use different types of quality products based on their requirements [5]. Every customer looks forward not for price but software quality because it saves resources in the account of software development time, maintenance costs, and efforts. So, considering software quality is being kept in mind by various researchers [7]. Soft Computing [8][9] based approaches have emerged as a very handy tool to go deep through the data-sets. There are well-known methods related to soft computing e.g. Neural Network based approach [10], Support Vector Machine [11], and Fuzzy logic [12]. The fuzzy logic consists of a fuzzy inference system that is employed to predict software quality based on the given inputs. The main advantage behind fuzzy logic is to expand intelligent machines and to resolve nonlinear and mathematical problems. This includes the calculation of a variety of factors determining the software quality, applying the obtained result to a fuzzy inference system or fuzzy system.

In this paper, we have discussed various factors by which software quality can be affected and found that five parameters namely Reliability, Efficiency, Usability, Maintainability, and Portability are sufficient for software quality estimation. Then, we proposed a fuzzy logic model to estimate the software quality level based on five parameters. The data-set to train the fuzzy inference engine has been collected as 243 rules by taking into account the expert's suggestions. The proposed methodology is validated where software quality is estimated automatically

with the proposed approach for given unknown data-sets and the results are compared mathematically with the actual quality.

The paper is organized as follows: Section II summarizes related work to predict software quality, Section III describes the factors used, Section IV elaborates the proposed FIS approach, Section V shows experimental design, Section VI presents the Experimental Results, Conclusion and Future Work is presented in section VII.

2. Related work

In order to measure quality one has to consider the factors that affect quality. Various researchers have taken into account different factors to measure quality and used different soft computing techniques to measure the level of quality. This section has discussed various standards of quality prediction techniques/methodology for predicting software quality using soft computing technique, namely Fuzzy Logic (FL).

Sangwan *et al.* [13] presented the existing models that are related the quality and used to predict the quality attributes in the software. Key characteristics considered are portability, maintainability, flexibility, usability, reliability and efficiency, and sub characteristics: accuracy, testability, extendibility, compatibility, understandability, and performance. It is observed that the selected set of quality attributes characteristics, Alvaro model stand out well in main characteristics as well as other sub characteristics namely understandability, accuracy and testability.

Mamta Punia *et al.* [14] proposed soft computing technique to automatically predict software maintainability levels, i.e. very good, good, medium, poor and very poor, and used fuzzy logic toolbox MATLAB to predict maintainability. Fuzzy logic is used to make set of rules and generating training and testing data sets. The performance was evaluated by using performance parameter Mean absolute relative error (MARE) and Mean relative error (MRE). Based upon experimental results, it is concluded that Artificial neural network (ANN) may be used to predict software maintainability level with reasonable accuracy.

Shaveta Gupta *et al.* [15] presented some soft computing techniques to computerize the software development process. Fuzzy model has been proposed and trained a fuzzy Inference system with four inputs: changeability, Interface complexity, understandability of software, and documentation quality to predict software reusability as output. Efficient use of soft computing techniques resulted in increased productivity and quality, reduced cycle time and lesser cost in the long run and found that to be suitable at the various phases of software development.

Kuldeep Singh Kaswan *et al.* [16] presented an overview of soft computing techniques and also compared software reliability modeling capabilities. Comparison outlined some parameters of modeling capabilities and observed that all the techniques explain its output and appropriate for complex model except genetic algorithm. It is also concluded that only fuzzy logic can be widely used for all modeling capabilities. Taboo search and cuckoo search can be used for most of the modeling capabilities except only small data set capability.

Charu Singh *et al.* [17] proposed a model based on five different factors that is Modularity (MD), Interface Complexity (IC), Maintainability (MN), Flexibility (FX) and Adaptability (AD) for the estimation of software reusability using soft computing technique through fuzzy logic. Proposed fuzzy model used some parameters as inputs which produces reusability as output. This is completed by assuming different membership functions such as Triangular, Trapezoidal and Gaussian membership functions defined in MATLAB to predict reusability. These all parameters into fuzzy sets: Less, Modder, More and the output reusability as Least, Less, Modder, More and Most and also used different membership functions such as Triangular Membership Functions (trimJ), Trapezoidal Membership Functions (trapmJ) and Gaussian guassmJ) Membership Functions defined in MATLAB for these parameters in order to predict the reusability and compared the result. This work would help estimate the reusability of these components with a possible accuracy, which will improve the software system's quality and estimate the development efforts.

Rudiger Lincke *et al.* [18] identified to significant differences among the quality models and experimental consist of context selection and hypothesis formulation. To represent suitable result, practitioners have to need a good understanding of software quality models, so apply them as black boxes. Evaluate a quality and compare these conclusions statistically to identify a significant differentiation between the qualities models therefore the selection of the quality estimation model has been based on software metrics.

3. Detailed analysis on various factors for quality assessment

There are so many software quality models which have been compared on the basis of various possible factors and a total of 35 factors are considered. 27 models for software quality assessment have been taken into consideration for a detailed analysis. The conventions of these models are denoted as follows and are given in Table I:

A- Mc Call‘s (1977)
B- Boehm’s (1978)
C- Murine (1983)
D- Bowen (1985)
E- Ghezzi (1991)
F- FURP’s (1987)
G- FURP+ (2000)
H- PERRY (1987)
I- Evans (1987)
J- Gilles (1987)
K- Detusch (1988)
L- ISO9126 (1991)
M- IEEE (1993)
N- AOSQUAMO (2009)
O- AOSQ (2012)
P- OOQM
Q- CSDQ
R- SQUARE’s (2011)
S- SATC’s
T- UML (2010)
U- QMOOD (2002)
V- KAZMAN (2003)
W- DEQUALITE (2009)
X- Dromey’s (1995)
Y- ISO25010 (2011)
Z- Bertoa
AA- Alavaro
BB- Account

S.N.	Attributes / Models	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	BB
1	Correctness	*		*	*				*	*	*	*																	7
2	Reliability	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*		*	*	*	*	25
3	Efficiency	*	*	*	*					*	*	*		*	*	*	*	*	*	*	*		*			*	*	*	19
4	Integrity	*		*	*	*			*	*	*	*									*					*			10
5	Usability	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	26
6	Maintainability	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*			*		*	*	*	*	23
7	Testability	*	*	*					*	*	*		*						*		*		*	*					11
8	Flexibility	*		*	*	*	*		*	*	*	*										*	*						11
9	Portability	*	*	*	*	*			*	*	*	*	*	*	*	*	*	*	*	*	*	*			*	*	*	*	22
10	Reusability	*		*	*	*			*	*	*	*	*									*		*	*				12
11	Interoperability	*		*	*				*	*	*	*	*									*				*			10
12	Human Engineering		*																										1
13	Understandability		*				*	*		*			*						*		*	*		*	*				10
14	Modifiability		*																		*	*		*		*			5
15	Functionality						*	*					*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	18
16	Performance						*	*					*							*					*	*			5
17	Supportability						*	*												*									3
18	Accuracy					*							*								*					*			4
19	Compatibility																									*			1
20	Feasibility																			*									1

21	Expandability				*					*		*									*		*				5
22	Manageability											*															1
23	Safety											*															1
24	Survivability				*							*															2
25	Verifiability				*					*		*															3
26	Modularity																				*		*				2
27	Evolveability														*												1
28	Modifiability		*																	*	*		*		*		5
29	Scalability																						*				1
30	Learnability																			*			*				2
31	Simplicity																						*				1
32	Security	*											*									*			*		4
33	Operability												*												*		2
34	Compatibility																								*		1
35	Availability	*	*			*	*	*					*	*				*	*	*	*		*		*			13

Where '*' shows that the particular software quality model includes this parameter

Here, based on the comparative study, we have identified five most important factors widely used by various researchers' which comprehensive software quality parameters namely: Reliability, Efficiency, Usability, Maintainability and Portability. The specifications of these factors are as follows:

Reliability: The ability to which the software product can sustain its level of performance under stated conditions for a stated period of time. For better quality reliability should be highly released.

Usability: The ability to which the software product makes it easy for users to operate and control it. For better quality usability should be highly released.

Efficiency: The ability to which the software product provides appropriate performance, relative to the amount of resources used, under stated conditions.

Maintainability: The ability to which the software product can be modified. In modifications included corrections, improvements or adaptations of the software to changes in the environment and in the requirements and functional specifications (the effort needed to be modified). For better quality maintainability should be less released.

Portability: The ability of the software product to be transferred from one environment to another. The environment may include extended hardware or software environment. The component should have high portability for better quality.

In the next section based on the five parameters we have proposed a FIS model to predict software quality.

4. Proposed FIS approach

In this section we proposed a fuzzy model using fuzzy system based on five parameters as input namely **Reliability, Efficiency, Usability, Maintainability** and **Portability** to estimate the software Quality. The proposed Fuzzy logic (FL) system is built, trained, and tested in MATLAB where following steps are followed in this approach.

Various parameters for the proposed approach are follows:

- First, Mamdani fuzzy model is chosen as a fuzzy inference system due to its simplicity and applicability. The proposed fuzzy model is shown in Fig.2 with five inputs and one output.

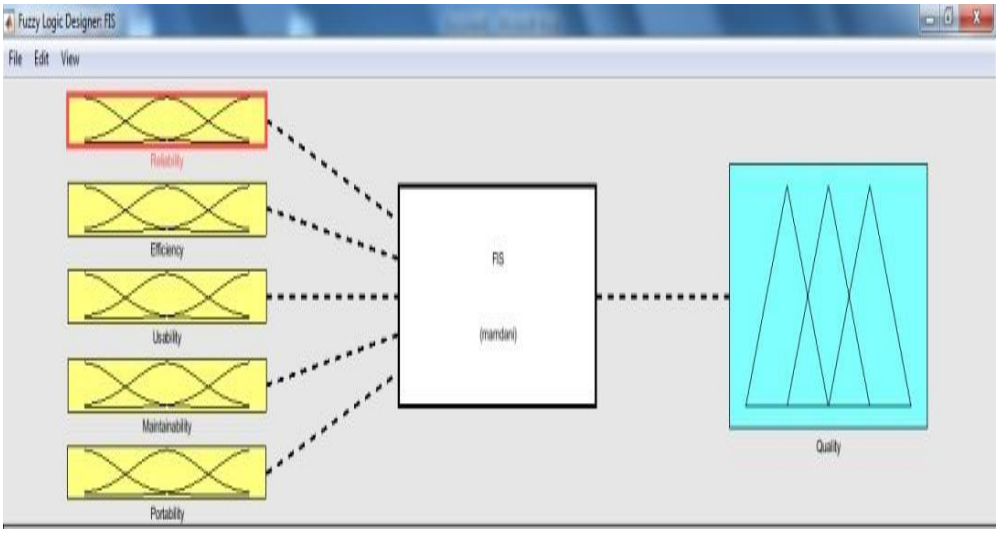


Fig.2. Proposed Software Quality Model with 5 inputs, 1 output

- Based on various studies in section 3, five parameter namely Reliability, Efficiency, Usability, Maintainability and Portability to access the software quality. Each parameter value lies in the interval [0 1].
- Each input parameter is divided into three fuzzy regions namely 'Low', 'Medium', 'High' and each fuzzy region employs Triangular membership functions as shown in the Fig.3.

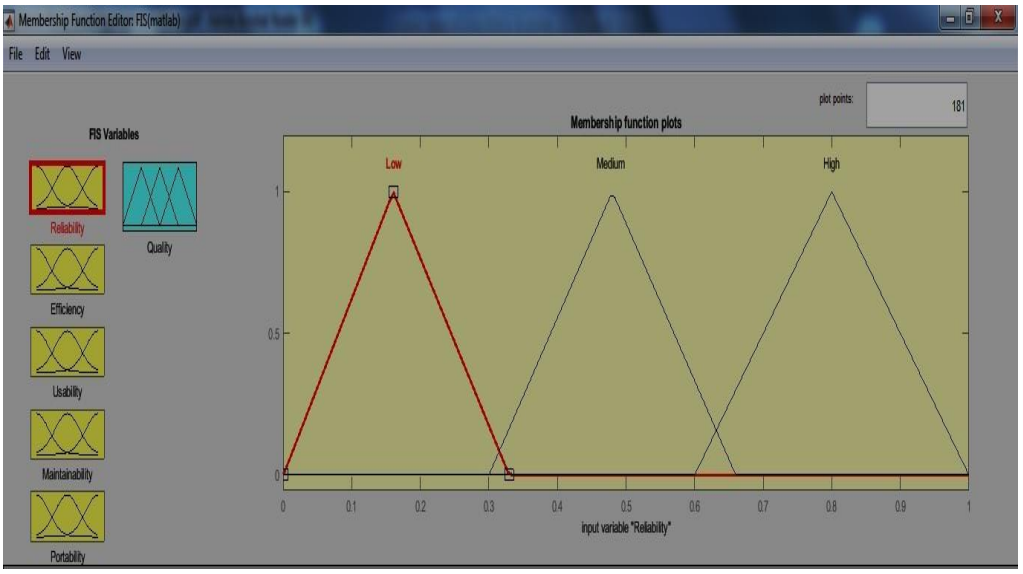


Fig.3. Membership Function Editor for Input Variable

- A total of 243 fuzzy rules are generated based on the various studies while talking all combination of five input parameters while taking into account the expert's opinion for the consequential part of each rule. A rule editor for MATLAB is shown in Fig.4 where a rule can be added or modified.

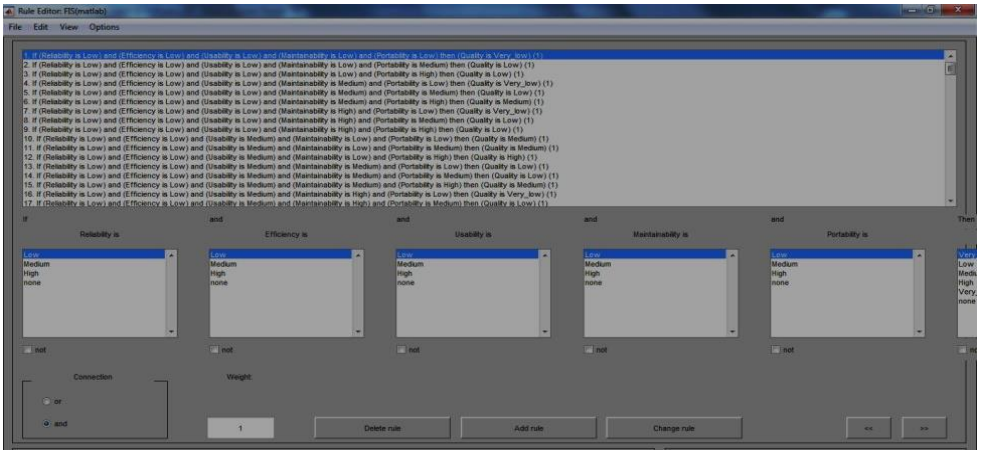


Fig.4. Rule Editor

- There is one output parameter namely 'Quality' of software which consists of five fuzzy regions where the quality of particular software lies. These fuzzy regions are 'very low', 'low', 'medium', 'high', and 'very high' as shown in the Fig5 . All these five fuzzy regions Triangular membership functions as well.

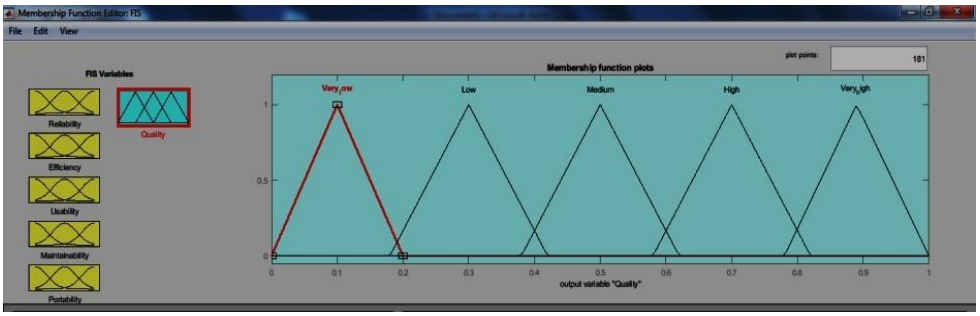


Fig. 5. Membership Function Editor for Output Variable

- We aggregated all the individual fuzzy sets for different rules and find output as a crisp value by defuzzifying the aggregated fuzzy set.

After selecting these parameters, fuzzy inference system is trained for all the specified rules. The overall FIS Structure is shown in Fig.6 where the quality of software can be checked by employing the procedure as follows:

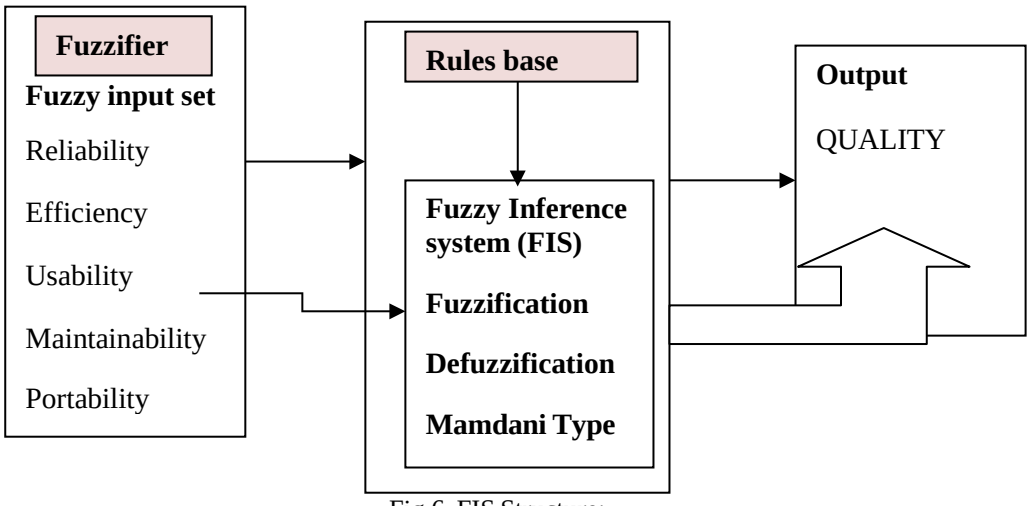


Fig.6. FIS Structure:

1. Collect five input parameters namely Reliability, Efficiency, Usability, Maintainability and Portability for the software which quality needs to be estimated and store these values.
2. Find the respective fuzzy region based on the each input parameters value for all the five input. Note down all the fuzzy regions covered by these inputs.
3. Write down all the possible combinations of the fuzzy regions obtained in step 2. Each combination takes one region at a time for a particular input.
4. Obtain the fuzzy rules which are satisfied with the combinations resulted in step 3. The only fuzzy rules match with step 3 combinations, fires and all other fuzzy rules becomes inactive.
5. Find the corresponding output fuzzy region based on the fuzzy rules which are fired.
6. Aggregate all output fuzzy region according to the 'Aggregation' method. We have selected 'max' aggregation method.
7. By employing any defuzzification method, we can evaluate the quality of the software. Defuzzification is a process in which fuzzy values are transformed into crisp quantities. Different methods used for defuzzification are centroid method, middle of maximum, smallest of maximum, largest of maximum, and weight average method etc. The most widely used method is centroid method and has massive applicability which is defined as:

$$qualqqty^{*} = \frac{\int x\mu_x(x)dx}{\int \mu_x(x)dx}$$

5. Experimental result and design

In proposed model for quality, the input variables taken are Reliability, Efficiency, Usability, maintainability and portability. Each of these consists of fuzzy sets low, medium, and high. We have chosen triangular membership function (trimf)and fuzzification process is applied on the above said sets by taking normalized values of the attributes given in table 1.First, we have implemented fuzzy inference system in which we have taken some values for Reliability, Efficiency, Usability, maintainability and portability as inputs and fuzzify these inputs based on the membership function namely triangular membership function (trimf) defined in MATLAB in order to predict software quality as output which is obtained by defuzzify the aggregated rules.

For experiment purpose we have created 243 rules along with dataset and 5 inputs whereas single output. We have used Mamdani-type model for our proposed Fuzzy Inference System. Crisp value for both input and output is defined as input variable membership function consists of (low, medium, high) and the output variable membership functions is quality, consisting of five fuzzy sets (very low, low, medium, high, and very high). The dataset for quality is referred as training data and testing data. Upon training, the training error has been shown by the FIS which reflect how good quality is achieved. Testing data has been used to confirm the model and to check how FIS behaves for known data. Fuzzy rules are generated in Mamdani fuzzy inference system (FIS) on the origin of input variables and membership functions by taking into account the expert's opinion for the consequential part of each rule. These rules are inserted to rule editor of fuzzy inference system as shown in figure 4. We aggregate all the individual fuzzy sets for different rules and find output a crisp value by defuzzifying the aggregated fuzzy set. After adding up of rules, we go to the view and select rule option. The Arise Rule Viewer window appeared which shows defuzzification of rules as shown in Fig.7.

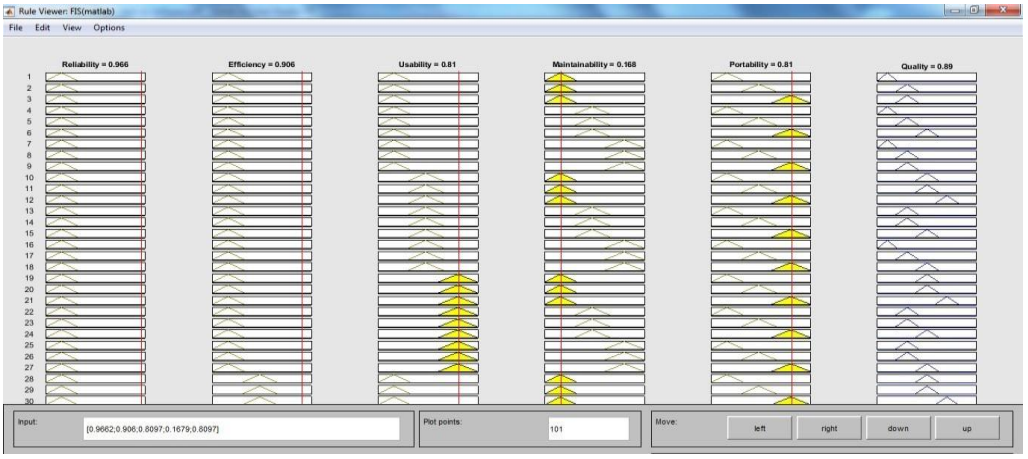


Fig.7. Rule Viewer

The proposed approach has been validated for various input data and quality is predicted. The obtained results for five sets of inputs are shown in Table 2 where predicted quality matches with the actual quality of the software.

Rules	**INPUTS**					**OUTPUT**
	Reliability	**Efficiency**	**Usability**	**Maintainability**	**Portability**	**Quality**
Rule.7	0.252(L)	0.267(L)	0.22(L)	0.69(H)	0.19(L)	0.1(VL)
Rule.8	0.297(L)	0.312(L)	0.28(L)	0.757(H)	0.504(M)	0.3(L)
Rule.123	0.59(M)	0.598(M)	0.586(M)	0.541(M)	0.78(H)	0.5(M)
Rule. 235	0.883(H)	0.861 (H)	0.832(H)	0.175(L)	0.183(L)	0.7(H)
Rule. 237	0.929(H)	0.914 (H)	0.966 (H)	0.138(L)	0.832(H)	0.89(VH)

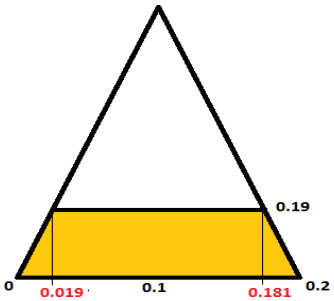
Table2. Experimental Result (Quality obtained for different values)

As shown in table 2, the values of quality are achieved 0.1, 0.3, 0.5 and 0.7 after giving the input values in theform of Reliability, Efficiency, Usability, Maintainability and Portability variables.

A detailed procedure is also provided below to obtain the software quality analytically. Consider first input combination from Table 2 as input values are given as **Reliability** is 0.252, **Efficiency** is 0.267, **Usability is 0.22, Maintainability** is 0.69 and **Portability** is 0.19. The goal is to estimate software quality based on these five input values. For this the same procedure is followed as given in section 4.

One fuzzy region is covered by each of these parameters values. The fuzzy region for these parameters are 'Low', 'Low', 'Low', 'High' , 'Low' for all five parameters respectively.
Then, we need to find the firing rules for this region combination. Only one rule e.g. rule 3 is fired in this case which says that the 'Quality' is 'Very Low'.

'Min' implication is employed for estimating rule output. Minimum of "**Reliability**= 0.252, **Efficiency** = 0.267, **Usability = 0.22, Maintainability** = 0.69 and **Portability** =0.19" is 0.19. Thus the fuzzified output of this rule can be shown as below:



Further, we have intended Quality value defuzzification with the help of centroid defuzzification method which is also called Center of Gravity or Center of Area.

$$quality^* = \frac{\int_0^{0.2} x\mu_x(x)dx}{\int_0^{0.2} \mu_x(x)dx}$$

$$quality^* = \frac{\int_0^{0.019} x.(10x)dx + \int_{0.019}^{0.181} x.(0.19)dx + \int_{0.181}^{0.2} x.(-10x+2)dx}{\int_0^{0.019} 10xdx + \int_{0.019}^{0.181} 0.19dx + \int_{0.181}^{0.2} (-10x+2)dx}$$

$$quality^* = \frac{\int_0^{0.019} x.(10x)dx + \int_{0.019}^{0.181} x.(0.19)dx + \int_{0.181}^{0.2} x.(-10x+2)dx}{\int_0^{0.019} 10xdx + \int_{0.019}^{0.181} 0.19dx + \int_{0.181}^{0.2} (-10x+2)dx}$$

$$= \frac{0.000022863 + 0.0031 + 0.00033814}{0.0018 + 0.0308 + 0.0018} = \frac{0.0035}{0.0344}$$

$$quality^* = 0.1017$$

The actual value of quality is 0.1. We see that the predicted value by proposed approach closely matches with the actual value.

Above experimental result showed that for better quality, its Reliability, Efficiency, Usability, and portability should be high whereas maintainability should be low.

6. Conclusion and Future work

Software Quality is additional and more significant in today marketplace. The quality models are used to predict the quality of software and use a different type of quality products based on user requirements. It gives a visual picture of software quality prediction models and found that a number of software quality models have been designed or evaluated. We have critically gone through the work done by various researchers to predict software quality models namely Mc Call, Bohem, and ISO etc to predict quality of software. It was found out that different researchers have taken different quality attributes to design/develop software quality models.

We have identified thirty four different software quality attributes considered by various researchers from twenty seven software quality models. Based on our analysis we have identified 34 distinct attributes/characteristics among 27 software quality models.

The main objectives are proposed models to reduce the time and increase productivity increase profit, improve marketing, increase customer satisfaction, and create a more useful and effective software quality. Finally, based on our work's factors, we applied fuzzy logic to develop a software quality model further to predict the level of software quality. Soft computing techniques play a vital role in developing software engineering applications. Besides its usage, other soft computing techniques will also be explained to design/develop a quality model to predict software quality. We have proposed a fuzzy logic model to estimate software quality level on the basis of five parameters: Reliability, Efficiency, Usability, Maintainability and Portability**.**

Our aim is to estimate quality of software because it has been widely traditional in both industry as well as academia and can help with both cost and time saving by employing the right choice of the software where the proposed approach can help.

References

1. Anju Gautam, Dr.vivek sharma, “An Integrated quality tools to evaluate the effectiveness of software”, World Academy of Informatics and Management Sciences, Vol.2 Issue1, ISSN: 2278- 1315, WAIMS, 2013.
2. Reiner R. Dumke, René Braungarten, Günter Büren, Alain Abran, Juan J. Cuadrado- Gallego, Software Process and Product Measurement: International Conferences IWSM Springer 2008.
3. UrvashiChauhan, ShrddhaSagar, “Analysis of Aspect Oriented Software Quality (AOSQ) Model”, Vol. 3, Issue 2, IJARCST 2015
4. NamitaMalhotra and ShefaliPruthi, “An Efficient Software Quality Models for Safety and Resilience”, IJRTE, ISSN: 2277-3878, Volume-1, Issue-3, 2012.
5. Y. Wang, and Yingxu Wang, “Software engineering standards: Review and Perspective”,Handbook of Software Engineering and Knowledge Engineering Chapter Id.
6. KavitaSheoran and Om PrakashSangwan, “An Insight of Software Quality Models Applied in Predicting Software Quality Attributes”, A Comparative Analysis, 978-1-4673-7231-2, IEEE2015
7. Garima Verma, Pradeep Tomar, “Predicting Quality using Fuzzy Model on Object Oriented SoftwareDesign,” The IUP Journal of Computer Sciences, Vol. 7, No. 4, pp. 18-29, 2013.
8. Shaveta Gupta and Jimmy Singla “Various Analysis and Design Techniques for Software EngineeringModel using Soft Computing”, IJCSCE 2013.
9. Kuldeep Singh Kaswan, SunitaChoudhary and Kapil Sharma, “Software Reliability Modeling using Soft Computing Techniques: Critical Review”, JITSE.volume 5, Issue 1, 1000144 ISSN: 2165-7866 page no. 90-101, 2015.
10. Thwin, Mie Mie Thet, and Tong-Seng Quah. "Application of neural networks for software quality prediction using object-oriented metrics." Journal of systems and software 76.2 (2005): 147-156.
11. [11 based on support vector machine." 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05). IEEE, 2005.
12. Jagatsesh, Arindampaul, “Quantification of Software Quality Parameters Using Fuzzy Multi Criteria Approach”, 978-1-61284-764-1, IEEE 2011.
13. KavitaSheoran and Om PrakashSangwan, “An Insight of Software Quality Models Applied in Predicting Software Quality Attributes”, A Comparative Analysis, 978-1-4673-7231-2, IEEE2015.
14. MamtaPunia and Er.AmandeepKaur,“Software Maintainability Prediction using Soft Computing

Techniques", IJISET Vol 1 Issue 9, November 2014.

15. Shaveta Gupta and Jimmy Singla "Various Analysis and Design Techniques for Software EngineeringModel using Soft Computing", IJCSCE 2013.
16. Kuldeep Singh Kaswan, SunitaChoudhary and Kapil Sharma, "Software Reliability Modeling using Soft Computing Techniques: Critical Review", JITSE.volume 5, Issue 1, 1000144 ISSN: 2165-7866 page no. 90-101, 2015.
17. Charu Singh, AmrendraPratap, AbhishekSinghal, "An Estimation of Software Reusability using FuzzyLogic Technique", 978-1-4799-3140-8, 2014 IEEE.
18. RudigerLincke,TobiasGutzmann and Welf Lowe, "Software quality prediction models compared", 1550-6002, IEEE 2010.
19. UrvashiChauhan, ShrddhaSagar, "Analysis of Aspect Oriented Software Quality (AOSQ) Model", Vol. 3, Issue 2, IJARCST 2015.
20. Sanjay Kumar Dubey, SoumiGhosh, Ajay Rana, "Comparison of Software Quality Models: AnAnalytical Approach", IJETAE ISSN 2250-2459, Volume 2, Issue 2, February 2012.
21. Ranbireshwar S. Jamwal,"Comparative Analysis of Different Software Quality Models", INDIACom- 2009.
22. Divya Prasad Naragani and PoonamUniyal, "Comparative Analysis of Software Quality Models", Vol2 Issue, ISSN 2278-733X March 2013.