INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 2)
Available online at: www.ijariit.com

Analysis of object-oriented system quality model using soft computing techniques

Anil Kumar
samt3410@gmail.com
Saroj Institute of Technology and Management, Lucknow, Uttar Pradesh

ABSTRACT

Quality plays a very important role in the software industry because the objective of every software industry is to produce good quality software with time and budget. A number of quality models have been proposed and used by various authors to build good quality software and these software quality models are also responsible for improving the performance, this improvement directly reflects the quality, increase users satisfaction and decrease the cost of quality. This report we have discussed and compared various quality models and soft computing techniques used for predicting the quality of software product and software system. It is found from the literature that there are lots of quality models and we measure quality based on their characteristics, so here, in this work, we give the characteristics definition and then a comparison of quality models related to quality characteristics. In this study, we exploring the information about soft computing technique for predicting the quality of a system or product and also develop a new quality model using soft computing (Fuzzy, Neural Network and Neuro-fuzzy) approach which is responsible to predict the software quality of an object-oriented system. In this report, we also identify the most important factors of an object-oriented system like Efficiency, Reliability, Reusability and Maintainability and also proposed a model based on these four factors that evaluating the quality of object-oriented system using soft computing technique i.e. Fuzzy Logic, Neural Network and Neuro-Fuzzy.

Keywords— *Software quality model, Soft computing techniques, Neural Network, Fuzzy Logic*

1. INTRODUCTION

1.1. Software quality

Quality plays a very important role in the software industry because the objective of the software industry is to produce good quality software with time and budget. So, here quality is nothing but the user's satisfaction about the product, suitable for the target & you can say confirmation of requirements. In a broad sense, user views of quality must deal with installation, Operational efficiency and convenience. Software's, Quality is commonly recognized as "Lack of bugs" in the Program.

1.2. Software quality model

Software quality models are a well-accepted means to support quality management of software systems. Over the last 30 years, a multitude of quality models have been proposed and applied with varying degrees of success. Quality models have become a well-accepted means to describe and manage software quality.

A software quality model is a tool for focusing on software enhancement efforts. As everyone knows that quality can be measured by many characteristics or attributes and to measure these quality characteristics requires several quality models and these software quality models are used for developing good quality software.

There are several software quality models for developing good quality software. So here we investigate some software quality models, description of quality models are described below and the diagram given below shows the classification of quality models (figure 1):

1.2.1. McCall's Quality Model: McCall's Quality Model is one of the most known quality models in software engineering literature.

1.2. Software quality models

Software quality models are a well-accepted means to support quality management of software systems.

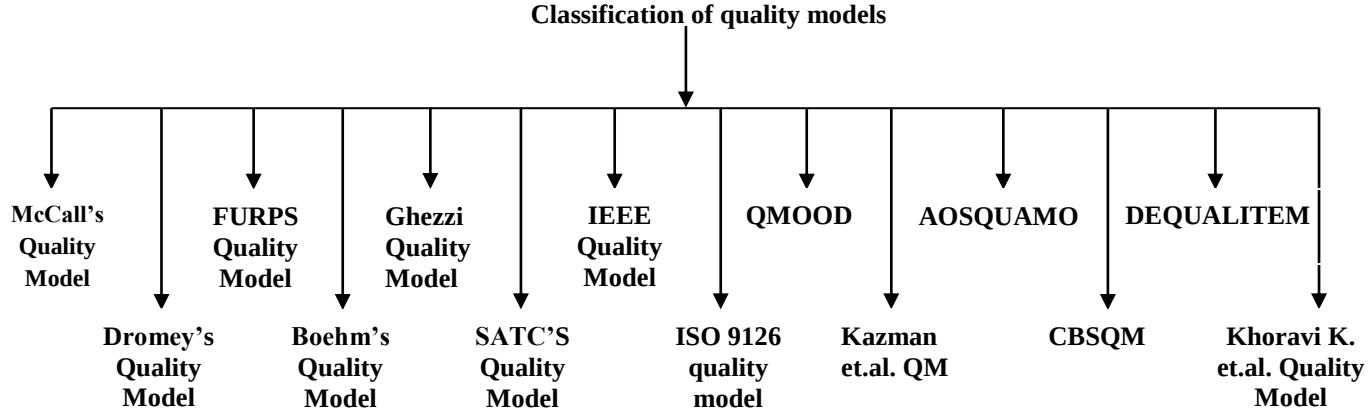

Fig. 1: Classification of quality models

1.2.2. Boehm's Quality Model: Boehm's quality model represents a hierarchical structure of characteristics, which describes that the purpose of the code is clear to the inspector but Architectural integrity is not covered in the model.

1.2.3. Dromey's Quality Model: This quality model has been presented by R. Geoff Dromey [4]. Who proposes a product based quality model which recognizes the quality evaluation differs from each product and also predict defects with the help of some violated properties.

1.2.4. FURPS Quality Model: as the basis of the name of the model is divided into 5 characteristics such as functionality, usability, reliability, performance and supportability which address the quality of software.FURPS model originally presented by Robert Grady and now IBM Rational Software extended it into FURPS+.

1.2.5. ISO 9126 Quality Model: The ISO 9126 quality model is the most useful none since it has been build based on an international consensus and agreement from all the country members of the ISO organization. ISO 9126 is an international standard for the evolution of software. The standard is divided into four parts which address respectively the following subjects: Quality model, External metrics, internal metrics and quality in use metrics.

1.2.6 Ghezzi Model: Ghezzi C. *et al*. state that internal qualities take care of the structure of software which provides framework for the software developers to get those external qualities for which software users concern and also provided Accuracy, Flexibility, Integrity, Maintainability, Reusability, Portability, Reliability and Usability characteristics which focuses both the internal and external qualities of software .

1.2.7 IEEE Model: IEEE is basically standard for software maintenance to provide a qualitative model. In this standard, an iterative process for management and execution of software maintenance activities has been described.

1.2.8 SATC's Quality Model: Software Assurance Technology Center (SATC) Hyatt L. E. *et al*. which is engaged for NASA with the objective of improving the software quality is actually helping the software managers in establishing metrics programs which may meet their basic needs with minimum costs and it is also interpreting the resulting metrics in the context of the supported projects.

1.2.9 QMOOD: A hierarchical Quality Model which extends the methodology of Dromey's quality model for Object-Oriented Design (QMOOD) proposed by Bansiya*et al.* [3]. It involves four levels as given below:

1. Identify characteristics of Design Quality
2. Identify properties of Object-Oriented Design
3. Identify the metric of Object-Oriented Design
4. Identify properties of Object-Oriented Inter Design

1.2.10 Component-based software development quality model: Sharma A. et.al, proposed a new model i.e. component-based software development quality model.

1.2.11. Kazman et al quality model: In this model author represented two new ideas about the quality characteristics through the software life cycle.

2. STUDY AND REVIEW OF LITERATURE

Kavita Sharma and Kumud Sharma [1], Software metrics and quality models play an important role in the measurement of software quality. There is various type of quality models which is responsible to build highly qualitative software. In this paper, author discuss four types of quality models consists of their number of characteristics.

Dr DeepshikhaJamwal [2], stated that different researchers have proposed different software quality models to help measure the quality of software products. Here in this research paper author discussed and compared the five quality models i.e. McCall‟s Quality Model, Boehm‟s Quality Model, Dromey's Quality Model, FURPS Quality Model and ISO 9126 Quality Model.

Sanjay Kumar Dubey *et al.* [3], with the idea of determining the multi-dimensional content in a more exact pattern various qualitative models have been presented by virtue of which different aspects of this matter have been attempted to be investigated properly. This paper may help to understand the purpose of a reference for investigating software quality and its related model.

JieXu, Danny Ho and Luiz Fernando Capretz, [5], states that software quality assurance plays a very important role in the software industry from the last few decades. Factors which characterize software quality can be identified because they may provide more importance for better software development and management. For achieving good quality in our software product we have to estimate quality at the early stage of our software product.

Brijendra Singh and Suresh Prasad Kannojia [11] states that if we use expression "software quality" then we think we got excellent software product that fulfils our expectations on the basis of basic components as constituent part of program or software and proposed a software quality prediction model based on the basic component.

Sana Shafi *et al*. [12] states that many quality prediction techniques are available in the literature to the prediction of the quality of software. However literature lakhs a comparative study to evaluate and compare various prediction methodologies so that quality professionals may select an appropriate predictor.

2.3 Discussion

In this report, a detailed analysis of software quality, software quality models, their characteristics, and the quality prediction of that model based on different techniques : machine learning based approach, fuzzy neural network based approach, neuro-fuzzy approach, basic component-based approach, conventional approaches and Fuzzy Expert-Systems generated by Genetic Algorithms based approaches are discussed. These approaches are responsible for effective software quality prediction which is helpful to provide an accurate result within budget and cost. The comparison of the approaches gives an answer to the effectiveness and efficiency of a Soft- Computing approach.

3. METHODS AND MODELS USED IN PROPOSED WORK

3.1. Introduction

This chapter gives the description of the study of methods, methodology and models which have included in proposed work. Here this section will discuss the details of the proposed problem and gives the procedure to implement it.

3.2. The methodology used in the proposed work

The methodology is a systematic and simpler way to solve our proposed problem. In this process, we will study the various steps that are generally adopted by the researcher in studying and understanding the proposed problem along with the logic behind them. In this figure given below show the working structure for estimating the software quality of an object-oriented system using soft computing techniques.

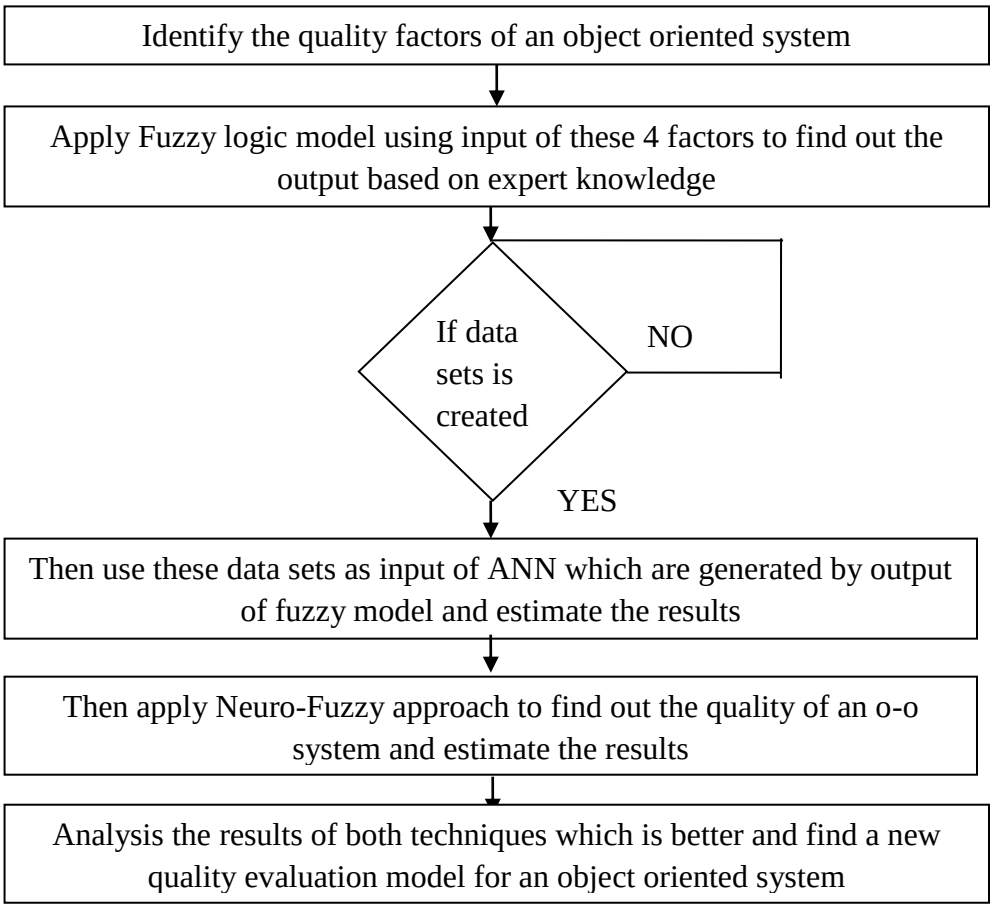

Fig. 2: Working methodology of proposed work

3.3. Tools and model used in proposed work

MATLAB - A tool used in soft computing techniques

MATLAB (matrix laboratory) is a fourth-generation programming language along with a numerical computing environment Developed by Math Works. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, the creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

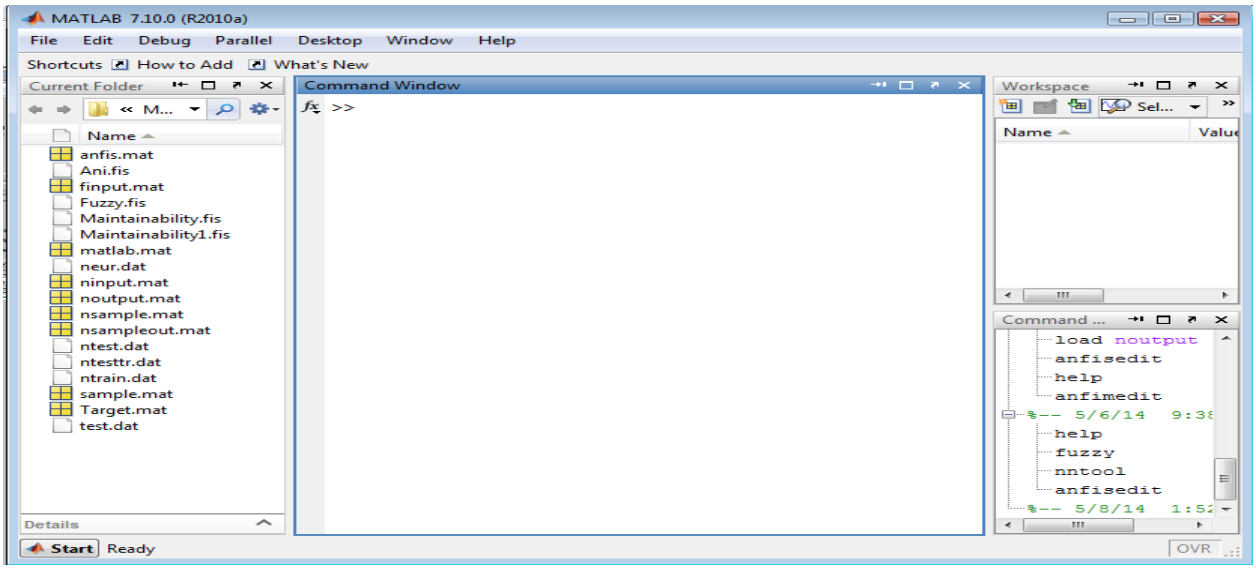

Fig. 3: a block diagram of Matlab

Here inside the MATLAB tool, we use more toolkits that are given below:

(a) Fuzzy logic toolkit: Fuzzy Logic Toolbox provides functions, apps, and a Simulink block for analyzing, designing, and simulating systems based on fuzzy logic. The product guides you through the steps of designing fuzzy inference systems. Functions are provided for many common methods, including fuzzy clustering and adaptive neuro-fuzzy learning. A simple block diagram of the fuzzy logic toolkit is given below in figure 4.

Fuzzy logic starts with the theory of a fuzzy set. A fuzzy set is a set without a crisp, visibly defined boundary. It can contain elements with only a partial degree of membership. In fuzzy logic, the truth of any statement becomes a matter of degree. A Membership Function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is time to time referred to as the universe of discourse, a imagine name for a simple concept.

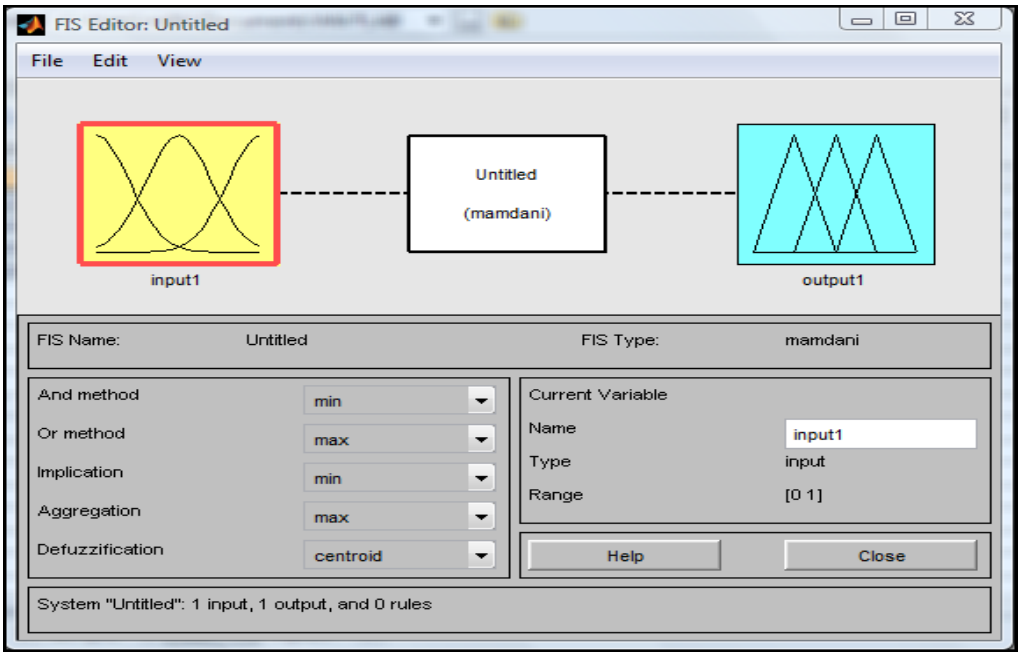

Fig. 4: Simple block diagram of the fuzzy tool

If this work is talk about the object-oriented system there are also some others factors which are good for predicting the quality of system so here this work have taken four input factors i.e. efficiency, reusability, reliability & maintainability because these quality factors are responsible to produce good quality of the system. This work assumes that software project management is fully concerned with the quality of the target system or software.

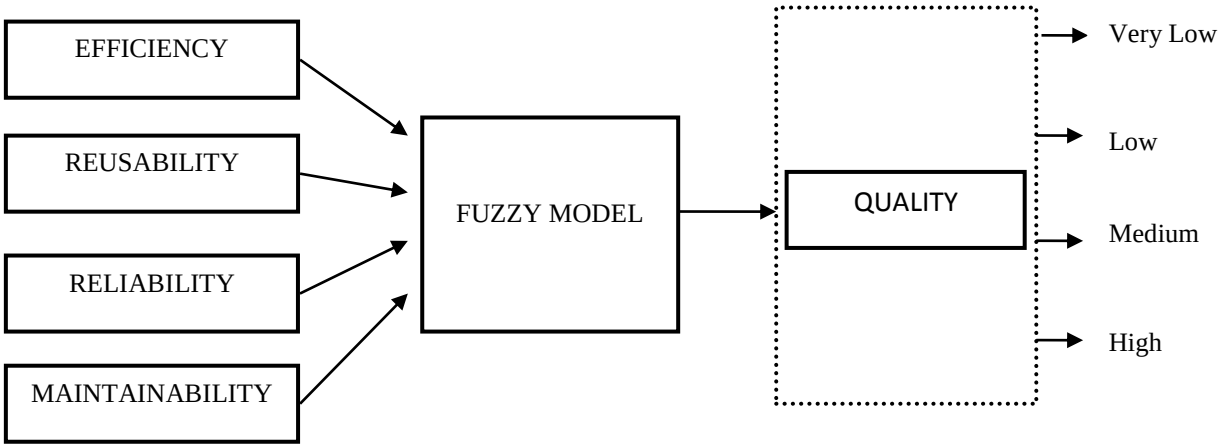

Fig. 4: Model of fuzzy logic

(b) Neural Network Toolkit (NNTOOL): The main element of an Artificial Neural Network approach is the original structure of the information processing system. It is the composition of a large amount of highly interconnected processing units called neurons working in union to solve specific problems. ANNs work just like people, gain knowledge (learn) by example. An ANN is configured for a particular application, such as data classification or pattern recognition, all the way through a learning process. In the biological systems learning process involves adjustments to the semantic connections that exist between the neurons.

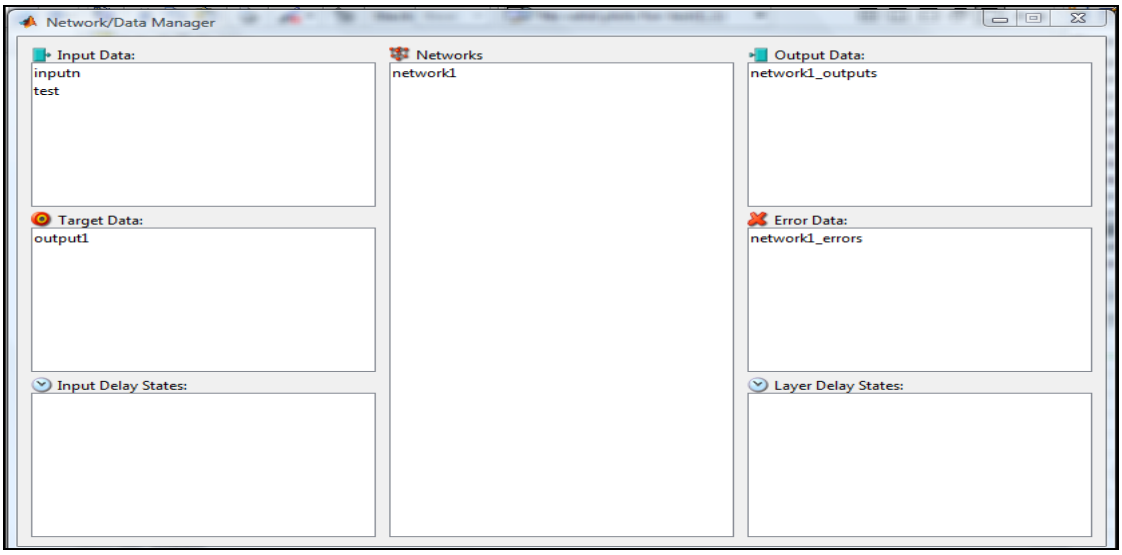

Fig. 5: Block Diagram of Neural Network

(c) Adaptive Neuro-Fuzzy Toolkit (ANFIS): Build Adaptive Neuro-Fuzzy Inference Systems (ANFIS), train fuzzy systems using neuro-adaptive learning. The basic idea behind these neuro-adaptive learning techniques is very simple: These techniques provide a method for the fuzzy modelling procedure to learn information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data.

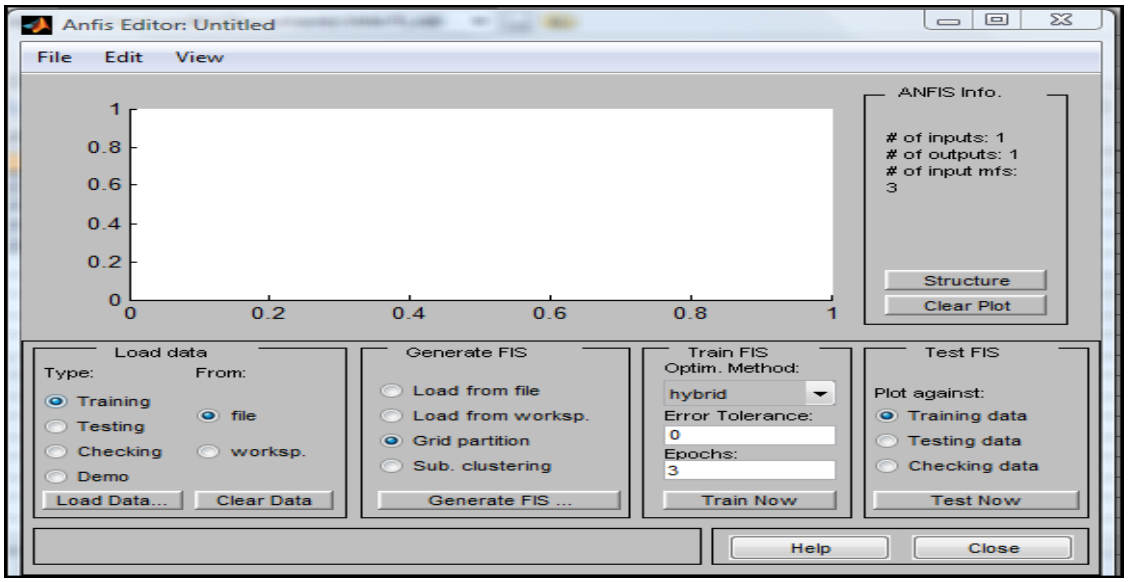

Fig. 6: a block diagram of Adaptive Neuro-Fuzzy editor

In this section gives the systematic process to achieve the proposed objective and also concluded the whole description of the tool and model which this work will be used.

4. PROPOSED WORK FOR ESTIMATING SOFTWARE QUALITY

4.1. Introduction

This chapter gives the full description of the whole proposed work according to our proposed objectives. Here, in this section, we will follow a systematic way to achieve the proposed objectives. This chapter also describes how to find out the solution of our Problem outline and also provides appropriate result according to the proposed objective.

4.2. Proposed work according to proposed objectives

Here this section will give the whole procedure how we will achieve our proposed objectives and also find out the solution of that objective.

In the diagram given below, we use phases of Rapid Application Development model instead of the software development life cycle phases of a basic waterfall model or **linear-sequential life cycle model.**

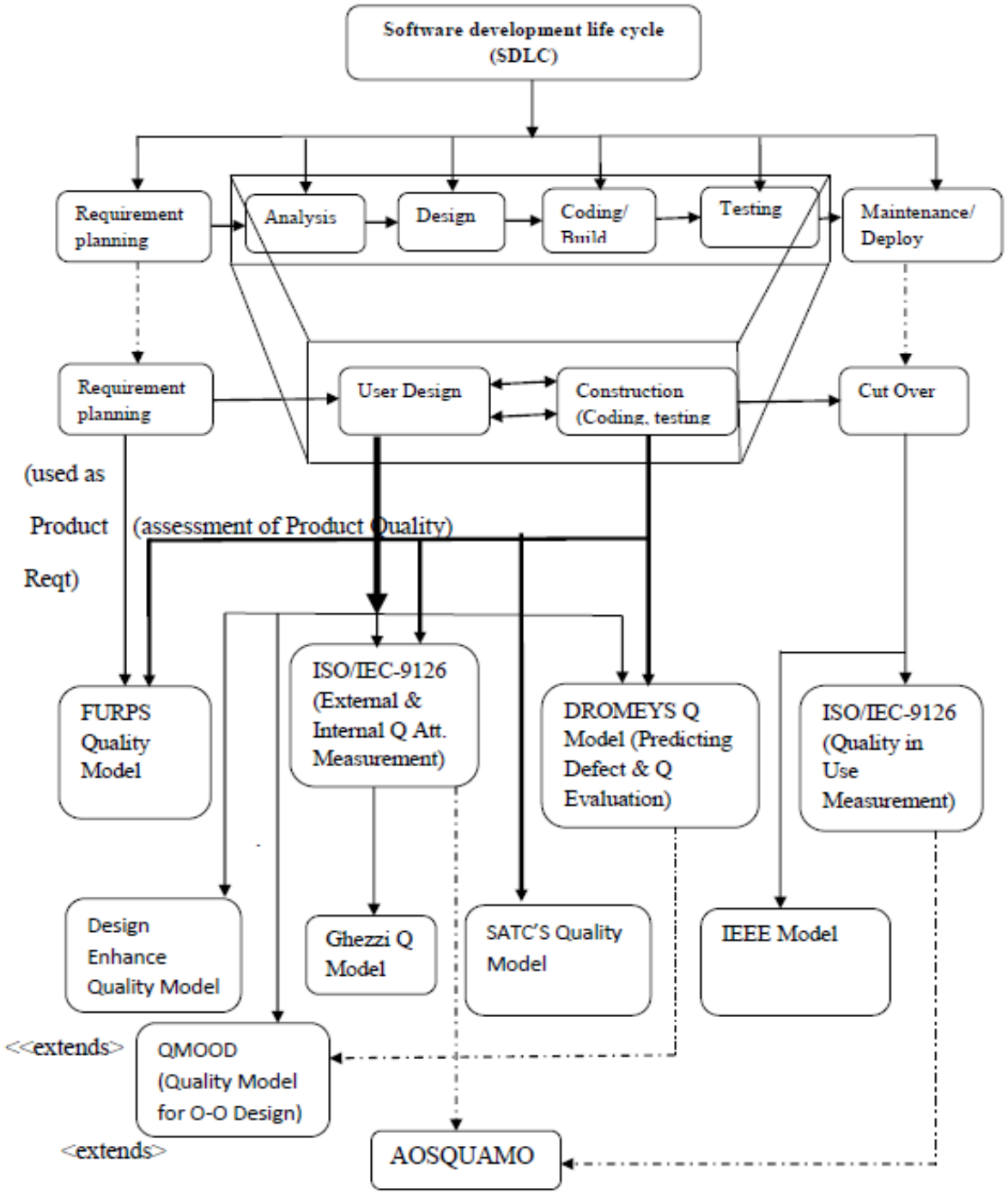

Fig. 7: Classification of Quality Models according to SDLC phases[a]

4.3 Summary of the first objective

Finally, in this section, we have concluded the objective 1 i.e. We classified various quality models according to the phases of software development life cycle by which we will save time and money for selecting the appropriate model for appropriate work.

Table 1: Difference between quality models based on quality factors

Quality Factors	Mc Calls	Boehm	FURPS	IEEE	Dromeys	Gheezi et.al	ISO- 9126	QMOOD	Kazman et.al	Kosravi K. et.al	AOSQUAMO
Accuracy		**Y**		**Y**		**Y**	**Y**				
Availability			**Y**	**Y**					**Y**		
Correctness	Y			**Y**							
Efficiency	Y	**Y**	**Y**	**Y**	**Y**		**Y**		**Y**		**Y**
Extendibility				**Y**				**Y**			
Flexibility	**Y**					**Y**		**Y**		**Y**	
Functionality			**Y**	**Y**	**Y**		**Y**	**Y**	**Y**		**Y**
Inheritability									**Y**		
Integrity	**Y**	**Y**				**Y**					
Interoperability	**Y**			**Y**			**Y**				
Maintainability	**Y**	**Y**	**Y**		**Y**	**Y**	**Y**				**Y**
Modifiability		**Y**							**Y**		
Modularity										**Y**	**Y**
Performance			**Y**								
Portability	**Y**	**Y**		**Y**	**Y**	**Y**	**Y**		**Y**		
Reliability	Y	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**				
Reusability	**Y**			**Y**	**Y**	**Y**		**Y**	**Y**	**Y**	**Y**
Robustness										**Y**	
Scalability										**Y**	
Security			**Y**	**Y**			**Y**		**Y**		
Supportability			**Y**	**Y**							
Testability	**Y**	**Y**	**Y**	**Y**			**Y**		**Y**		
Understandability		**Y**					**Y**	**Y**		**Y**	
Usability	**Y**		**Y**	**Y**		**Y**	**Y**			**Y**	**Y**

Table 2: rating of quality factors

S.no	Quality Factors	Rating of each quality factors according to their use in various models based on the table-1
1.	Accuracy	**4**
2.	Availability	**3**
3.	Correctness	**2**
4.	Efficiency	**8**
5.	Extendibility	**2**
6.	Flexibility	**4**
7.	Functionality	**7**
8.	Inheritability	**1**
9.	Integrity	**3**
10.	Interoperability	**3**
11.	Maintainability	**7**
12.	Modifiability	**2**
13.	Modularity	**2**
14.	Performance	**1**
15.	Portability	**7**
16.	Reliability	**7**
17.	Reusability	**8**
18.	Robustness	**1**
19.	Scalability	**1**
20.	Security	**4**
21.	Supportability	**2**
22.	Testability	**6**
23.	Understandability	**4**
24.	Usability	**7**

(a) Efficiency: Efficiency is "the capability of the software product to provide appropriate performance, relative to the number of resources used, under stated conditions" [36]. In general, it describes the degree to which time, effort or cost is well used for the intended task or purpose, which characterizes how efficiently the user can complete his task. They defined efficiency in a number of components as:

- Number of goals/task not achieved;
- Time is taken for task completion;
- Unproductive period; and
- Percentage of the task not completed.

(b) Reusability: Software reuse has widely used the development of software using some existing software workings. Software reuse has been used as a tool to decrease development cost and time. The ease with which an existing application or component can be reused. It is the key element to improve the quality of the system and reduce the cost. Reuse could save 20% of the development cost. Software reusability more purposely refers to design features of a software element or group of software elements that improve its appropriateness for reuse.

(c) Reliability: Reliability is the capability of the software product to maintain a specified level of performance when used under specified conditions. "Reliability is the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered."

(d) Maintainability: Maintainability is the most important factor of software quality. It is defined as the possibility of running a successful repair operation within a given time. According to IEEE, it is defined as: "The ease with which a software system or component can be modified to correct faults, improve the performance or other attributes, or adapt to a changed environment is maintainability".

4.4 Proposed fuzzy logic model approach

Fuzzy logic is a mathematical tool which provides a simple method to arrive at an exact conclusion based upon noisy, vague, imprecise, ambiguous, noisy, or missing input information. Fuzzy logic also offers a unique suitable way to create a mapping between input and output places by using the natural expression.

- Fuzzification: This is to find out the degree to which the input data match the condition of the fuzzy rules.
- Inference system- Fuzzy inference system is the technique of formulating the mapping from a given input to output using fuzzy logic. Based on its corresponding degree it calculates the rule's decision.
- Composition: To join the conclusion inferred by all fuzzy rules into a final conclusion.
- Defuzzification: To convert a fuzzy conclusion into a crisp one. The input for the defuzzification process is a fuzzy set and the output is a single number.

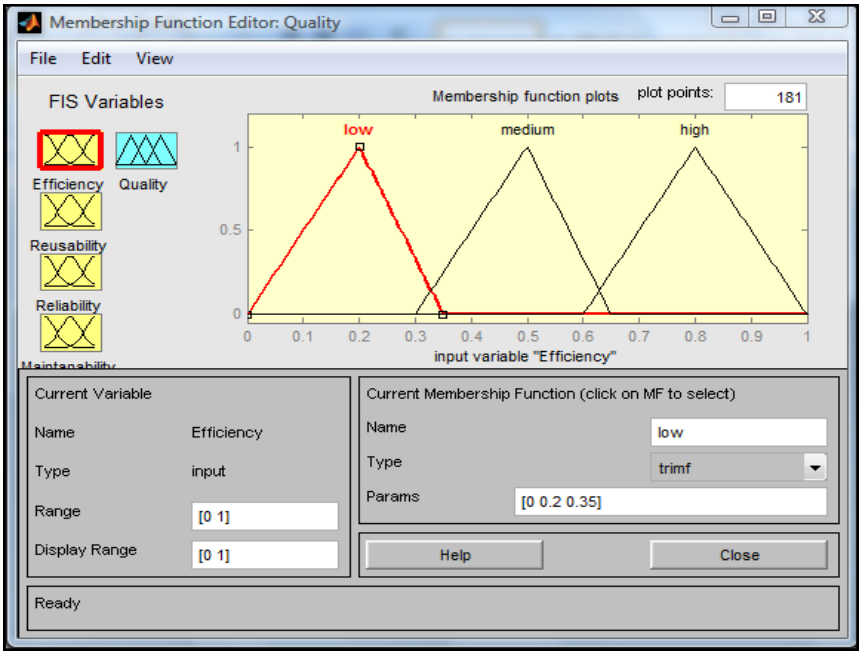

Fig. 8 (a): Membership function for efficiency

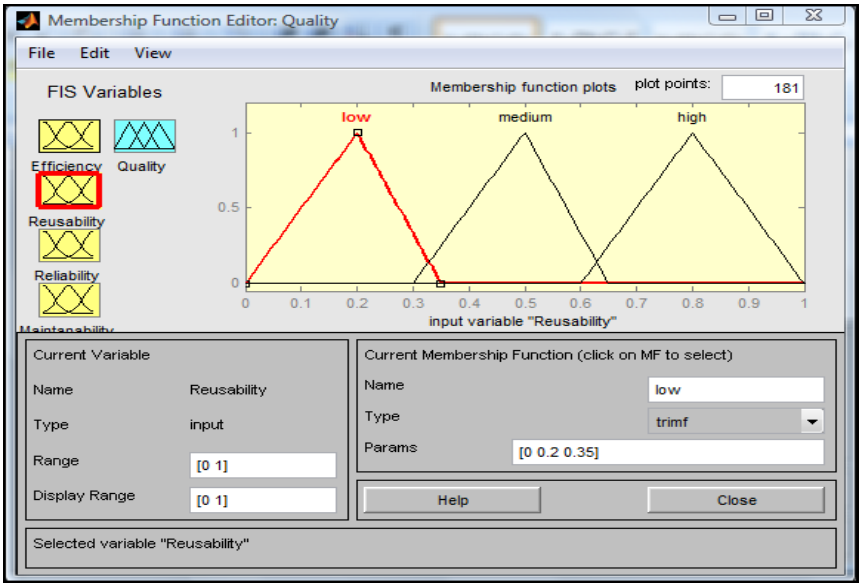

Fig. 9 (b): Membership function for reusability

4.5 Proposed Neural Network Approach

Neural networks have been trained to execute complex functions in various fields, including pattern recognition, classification, identification, vision, and speech and control systems. Data sets are generated by the fuzzy model and now work as input datasets to the proposed Neural Network Model. It is well-known that the Neural Network model can be used to model any arbitrary input-output mapping and are capable of approximating any measurable function.

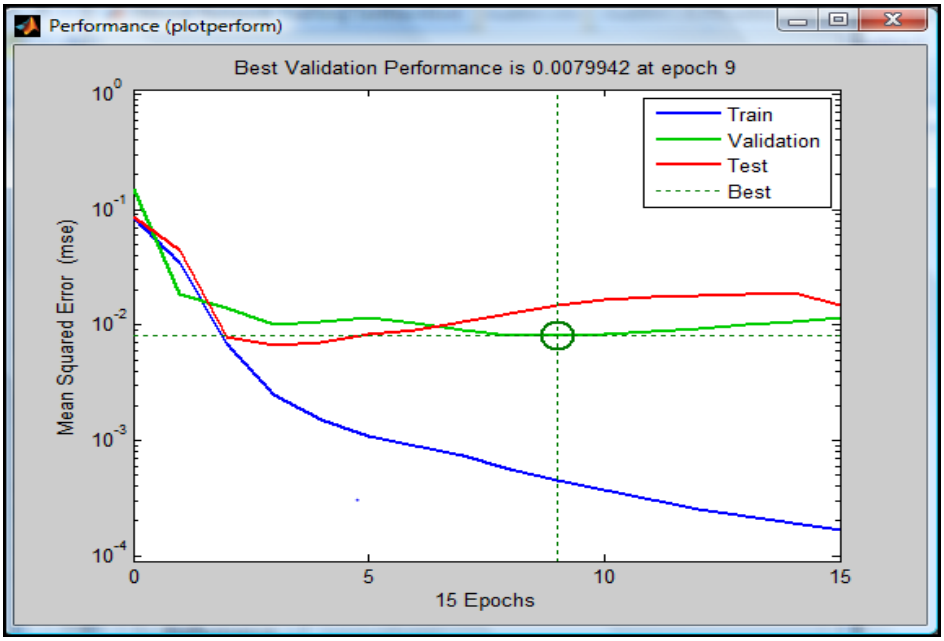

Fig. 10: Diagram of the Performance graph

Input Units	4
Output Unit	1
Hidden neurons	10
Output neurons	1
Training of Neural Network	
Algorithm	Back Propagation
Training Function	trainlm
Transfer Function	Tansig
Activation function	Pure linear

4.6 Proposed Neuro-Fuzzy Approach

The fuzzy logic approach is advantageous for measuring the quality of software components because the conventional model-based approaches are complicated to be implemented. Regrettably, with the increase in the complexity of the problem being modelled various constitutes for measuring the quality, has led to rely on another technique which is mostly known as the Neuro-Fuzzy approach.

Validation Results of ANN

The performance function used here is Mean Absolute Error (MRE), Mean Absolute Relative Error (MARE) and Root Mean Square Error (RMSE). The model is trained using training data sets and was evaluated on validation data sets. Table 3 shows the MRE, MARE and RMSE results of ANN model evaluated on validation data set.

Table 3: Validation Results of ANN Model

Performance measures	MRE	MARE	RMSE
Validation set using neural network	0.009088	0.010515	0.18574

Experimental result of the measured software quality of the validation set is shown in figure 11. From which it becomes recognizable that our proposed model produces Quality levels that are comparable to the measured quality levels as predicted by the proposed fuzzy model.

5. RESULT ANALYSIS

This section represents the analysis performed to find a relationship between independent input variables i.e. efficiency, reusability, reliability and maintainability and one dependable output variable i.e. software quality. The data set contains 110 different data sets generated by the rule base of fuzzy logic.

5.1 Validation results of ANN

The performance function used here is Mean Absolute Error (MRE), Mean Absolute Relative Error (MARE) and Root Mean Square Error (RMSE). The model is trained using training data sets and was evaluated on validation data sets. Table 4 shows the MRE, MARE and RMSE results of ANN model evaluated on validation data set.

Table 4: Validation Results of ANN Model

Performance measures	MRE	MARE	RMSE
Validation set using neural network	0.009088	0.010515	0.18574

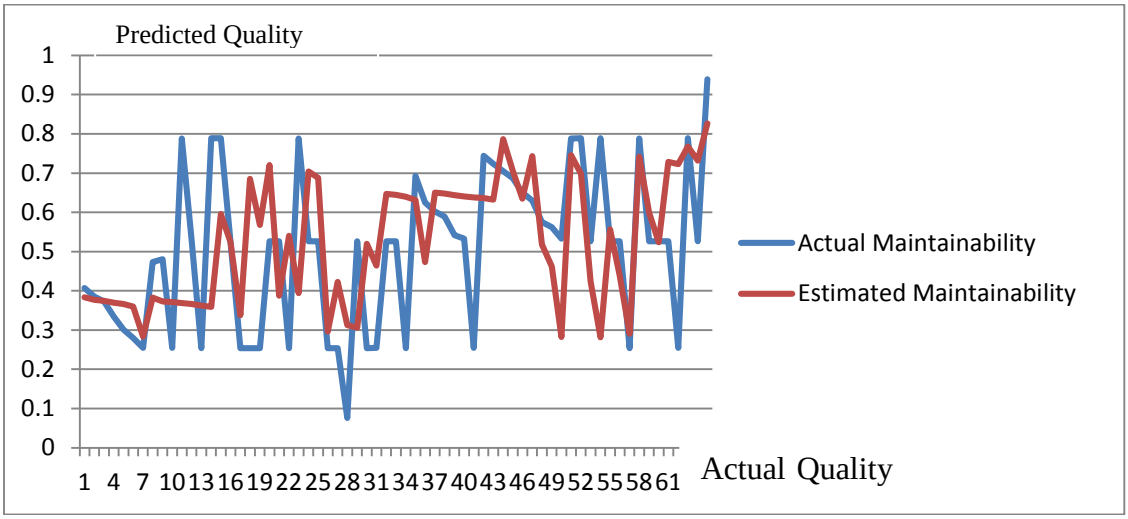

Fig. 11: Validation Result of ANN with 10 neurons

Table 5: Validation results of ANN Model and Neuro-fuzzy model

Performance measures	The validation set using ANN (at Training)	The validation set using neuro-fuzzy approach (at Training)
RMSE	0.18574	0.0016

While the actual output produced by the proposed "Neuro-Fuzzy Quality Evaluation System" is near to the expected output, therefore the proposed model may be suggested for automatic prediction of software quality level for the object-oriented system.

6. CONCLUSION AND FUTURE DIRECTIONS

This report contains the precise and exact description of the software quality characteristics and also provides a description of quality models. After that, it gives the classification of the quality model according to SDLC which is very useful for developers and users also. This work also gives knowledge of the quality prediction approach based on soft computing techniques i.e. fuzzy model, Neural Network and at last, the neuro-fuzzy technique allows the integration of mathematical data and expert knowledge. In the domain of Software Engineering, this technique can be made a powerful tool to handle necessary and useful problems. In this report, we have described each and every aspect of the proposed research problem. It comprises the definitions and gives an explanation of all the technical terms related to the report, setting out the objective of the report and specific scope. And hence this report is of great importance for the developers and the information provided in this document will also be important for the researchers for predicting the quality of a system. Apart from that this document will also guide and assist each and every student related to this work.

Future directions of this report are that this work can be estimated using Genetic Algorithm, Swarn Intelligence and Ant colony optimization with online available data sets and also estimated its result with the available techniques.

7. REFERENCES

[1] Kavita Sharma, Kumud Sharma (2013) "Comparison of Various Software Quality Models"Recent Trends In Computing and Communication Engineering -- RTCCE 2013
[2] Dr DeepshikhaJamwal (2010) "Analysis of Software Quality Models for Organizations" International Journal of Latest Trends in Computing 19Volume 1, Issue 2, December 2010
[3] Sanjay Kumar Dubey1, Soumi Ghosh2, Prof. (Dr.) Ajay Rana3 (2012) "comparison of software quality models: an analytical approach" International Journal of Emerging Technology and Advanced Engineering Volume 2, Issue 2, February 2012
[4] Rafa E. Al-Qutaish, PhD (2010), "Quality Models in Software Engineering Literature: An Analytical and Comparative Study"Journal of American Science; 6(3), 2010
[5] JieXu, ²Danny Ho and ¹Luiz Fernando Capretz" an empirical study on the procedure to derive software quality estimation models"International Journal of Computer Science & Information Technology (IJCSIT) Vol.2, No.4, August 2010
[6] Gaffney, J. E.1981." Metrics in software quality assurance", ACM press. Volume no. 81, pp.126-130, 1981
[7] Hyatt, L. E. and Rosenberg, L. H.1996. Product Assurance Symposium and Software Product Assurance Workshop, Proceedings of meetings, European Space Agency, pp. 209, 1996
[8] Sharma, A., Kumar, R. and Grover, P. S. 2008. "Estimation of Quality for software components: an empirical approach", ACM SIGSOFT Software Engineering Notes, 33, 2008
[9] Hyatt, L. E. and Rosenberg, L. H.1996. Product Assurance Symposium and Software Product Assurance Workshop, Proceedings of meetings, European Space Agency, pp. 209, 1996
[10] Brijendra Singh, Suresh Prasad Kannojia"A Model for Software Product Quality Prediction"Journal of Software Engineering and Applications, 5, 395-401, 2012

[11] Hamdi A. Al-Jamimi and Moataz Ahmed "Machine Learning-based Software Quality Hamdi A. Al-Jamimi and Moataz Ahmed "Machine Learning-based Software Quality Prediction Models: State of the Art" Information Science and Applications (ICISA), pp1 - 4, 2013
[12] Ekkehard Baisch and Thomas Liedtke" Comparison of conventional approaches and Soft-Computing approaches for Software Quality Prediction" 1997 IEEE
[13] Bo Yang et al "Early Software Quality Prediction Based on a Fuzzy Neural Network Model"Third International Conference on Natural Computation (ICNC 2007) Volume 01 Pages 760-764, 2007
[14] Trendowicz, A. and T. Punter. "Quality Modeling for Software Product Lines. In": 7thECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'03). 2003.
[15] Dromey, R.G., "A Model for Software Product Quality" IEEE Transactions on Software Engineering, 21(146-162), 1995
[16] Zhang, D. and J.J.P. Tsai, "Machine Learning and Software Engineering" Software Quality Journal, 11(2): p. 87-119, 2003
[17] K.A.De Jong, " Analysis of Behaviour of a class of Genetic Adaptive Systems".PhdThesis, University of Michigan, Ann Arbor, MI, 1975.
[18] The standard for Software Maintenance, Software Engineering Standards Subcommittee of the IEEE Computer Society IEEE 1993.
[19] Tong-SengQuah and Mie MieThetThwin. "Application of Neural Networks for Software Quality Prediction Using Object-Oriented Metrics": the International Conference on Software Maintenance (ICSM'03), 2003 IEEE.
[20] Cong Jin, Shu-Wei Jin, Jun-Min Ye and Qing-Guo Zhang "Quality Prediction Model of Object-Oriented Software System using Computational Intelligence"2009 2nd International Conference on Power Electronics and Intelligent Transportation System, 2009 IEEE.
[21] Wei PENG and Lan YAO "An Approach of Software Quality Prediction Based on Relationship Analysis and Prediction Model"2009 IEEE.
[22] Yajnaseni Dash and Sanjay Kumar Dubey"Quality Prediction in Object Oriented System by Using ANN: A Brief Survey"International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 2, February 2012.

