IJARSE
ISSN 2319 - 8354

Enhancing the Estimation of Reliability using Enhanced Adaptive Neuro-Fuzzy Inference System

Mr. P. Ramesh[1], Dr. Prasath S[2]

[1]Ph.D Research Scholar in School of Computer Science,
VET Institute of Arts and Science (Co-education) College, Thindal, Erode, Tamil Nadu, India
and Assistant Professor and Head, Department of Computer Science,
Kongu Arts and Science College (Autonomous), Erode, Tamil Nadu, India.
[2]Assistant Professor and Head, School of Computer Science,
VET Institute of Arts and Science (Co-education) College, Thindal, Erode, Tamil Nadu, India

ABSTRACT

Estimate the reliability at a premature stage assists to optimize the testing and safeguarding of the software. Quick alterations in hardware and software technologies direct to inventions of novel methodologies. To develop and validate the reliability estimating model using Machine Learning technique which is able to provide a solution for this problem. Software reliability is an imperative quality characteristic and there are a lot of models that try to estimate the reliability from different perspectives. Selecting the proper metrics is an important one to estimate the reliability of the system. In this paper, software is developed based on design of Service-Oriented Software System. In Service-Oriented Software System, the complete software system contains an interacting group of autonomous services. Hence, need to estimate the reliability with better performance. In this study, numerous reliability models were analyzed by a soft computing model called Enhanced Adaptive Neuro-Fuzzy Inference System (EANFIS), which is used to estimate the reliability of a component and the probability that a component carried out without fault. To estimate the coefficient of determination by the authenticate model.

Keywords: *Service-Oriented Software System, Neural Networks and Fuzzy Logic, EANFIS, Reliabilty.*

I. INTRODUCTION

Estimating the reliability of Software Development Life Cycle makes the software less costly and move forward its timely release. The essential actions of software development process are software testing and software maintenance. Many researches have been carrying out to reduce the cost but it is not compromise the quality and accuracy. To produce a reliable software product and ensure the reliability in maintenance process are the main objectives. Estimating the reliability of a product before testing be able to present expensive imminent about the qualified measures. It can on the whole improve the planning of testing and the maintenance process. Various estimation models of reliability for software like In [1], Rome Laboratory model, In [2] Putnam’s model and several models are used to estimate the software reliability. These models can estimate the reliability after the testing phase. A lot of approaches and methods in Data mining and Machine learning are used by the researchers. Prior to entering the testing phase, the development in algorithms and software computing methods are used for estimating the reliability of the software. In [3], extraordinary result was formed by Adaptive Neuro-Fuzzy Inference System compared with a range of machine learning methods.

Based on the programming paradigm the design metrics can measure the quality of software. By selecting an appropriate design metrics as an input, calculate the occurrences of fault in the software, recognize the probable residual fault, estimate the development cost, estimate the required time to release, and sketch resource utilization. In this paper, Estimation of reliability is measured by Service-Oriented Software System. A Service-Oriented Software System is prepared by loosely coupled components. These components were used in a distributed environment. Service-Oriented Software System is trendier paradigm. Service-Oriented Software System is compared with component-based system. In Service-Oriented Software System the components are loosely coupled but in component-based software system each individual component as a software package. These components are put together so that every component can communicate with another via their interfaces. This paper concentrates to estimate the reliability of the individual component. The failure pattern of the individual component depends on the design and coding of the system. The Section 2 shows the related work. Section 3 describes the service oriented software system. Section 4 discusses the neural networks and fuzzy logic. Section 5 defines the proposed methodology for predicting the reliability. In section 6, describes the results. Section 7 conclude the work.

II. RELATED WORK

Estimation of reliability is an important task in software engineering. Many researchers have taken dissimilar metrics to estimate the reliability. The factors which are used for estimation of reliability based on faults, estimation of reliability depends on the design metrics such as coupling and cohesion, estimation of reliability based on the difficulty of the software system. The parameter fault is a critical dimension to estimate the reliability of a component-based system. In [4], estimate the parameter fault in a component-based system using the Halstead Software Science. To make the system reliable to identify the faults before testing phase because faults can cause more damage at the testing stage. Take out the parameters from the unstructured textual requirement and developed a innovative model for fault prediction. This model identifies the faults which may cause failure in the system in [5]. Both coupling and cohesion are important factors in determining the maintainability, scalability, and reliability of a software system. Many researchers find that cohesion and coupling are the most important metrics to initiate the faults in the system. Higher coupling and lower cohesion creates the chances for fault induction and minimize the reliability of the system. In [6], measuring the software reliability to use static and dynamic cohesion and coupling metrics. Class cohesion or rate of association within a class in object oriented systems was crucial. High coupling and low cohesion can make a system difficult to change. In [7], the article introduces a software measurement model that uses a heuristic normalization of the software's internal quality attributes, i.e., coupling and cohesion, for software quality measurement. In addition, it identifies the locations in software design that exhibit unnecessary couplings that degrade the quality of the software systems, which can be eliminated. The complexity of cohesion and coupling makes the system more vulnerable which is discussed in [10]. While the complexity enlarges, then the faults in the software system get enlarged. The major concern to improve the reliability is to measure software complexity. Cyclomatic Complexity is a third reliability parameter which is talk about here. Cyclomatic complexity is software metric used in software development. Cyclomatic complexity can be calculated with respect to functions, modules, methods or classes within a program. Cyclomatic complexity can be calculated manually if the program is small. Automated tools need to be used if the program is very complex which was described in [11]. In [12] the potential of CC as a predictor of software quality and maintainability holds tremendous promise for significant breakthroughs in software development techniques. However, Cyclomatic Complexity is a critical software metric which quantifies the intricacy of a program structure and control flow.

III. SERVICE-ORIENTED SOFTWARE SYSTEM

Service-Oriented Software System gives a intend structure for system development and enhancing total system quality. Web services standards and technologies are used by Service-Oriented Software System. It provides a software concept based on services. Service-Oriented Software System transports a number of profits. They included as follows, It reduces the time to market, improves the business alignment for development, minimized the costs, reduce the business risks. Service-Oriented Software System acts as service provider, service broker or service register and service requestor or web service. The role of service provider is to make trade-offs between availability and security. The task of the service broker or service register is to make information for requestor. The job of service broker is to decide the quantity of information to transfer. The service requestor or Web service process the client requests for a service. It binds to the service provider to call upon one of its Web services.

Services were loosely coupled, autonomous, and reusable. They have well-defined platform-independent interfaces. Service is a realization of a well-defined business functionality that operates independent of the state of any other service defined within the system.

IV. NEURAL NETWORKS AND FUZZY LOGIC

Two basic elements of soft computing techniques are Neural Networks and Fuzzy Logic. Neural networks are a form of multiprocessor computer system. It has simple processing elements, a high degree of interconnection and an adaptive interaction between elements. It is called as an Artificial Neural Network (ANN). Many different kinds of learning rules are followed by neural networks. ANNs can learn from data and feedback and have learning capabilities.

Fuzzy systems are apt for estimated reasoning. Fuzzy logic allows decision making with estimated values. A fuzzy set is a generalization of an ordinary set by allowing a degree of membership for each element. The membership-function maps each element to its degree. A membership degree is a real number on 0 and 1. In extreme cases, if the degree is 0 the element does not belong to the set, and if 1 the element belongs hundred percent to the set.Fuzzy logic models are rule-based models and do not have learning capabilities. The operations of Fuzzy inference system are Fuzzification of the input variables, Determination of membership functions for the parameters, Application of the fuzzy operator in the predecessor, Implication from the precursor to the consequent and defuzzification.

V. PROPOSED METHODOLOGY

Software metrics measures the quality of software, certain properties and specifications. The software design metrics plays a significant role in the prediction of the fault. There is some metrics on which the reliability of the software may depend. From the expert opinion, the four software metrics such as Faults, Cohesion, Coupling, and Cyclomatic Complexity are identified to develop the architecture of proposed methodology.

The developer created the typing mistake is known as fault. It is an uncorrected step which discontinues the system to perform the selected task. Minimum faults produce the better reliability of the system. In the Software Development Life Cycle, the fault occurrence crates the problem from Requirement phase to the Development phase. An execution of fault produces failure of the system. Hence, premature detection of a One of the Metric which predicts the fault before the system undergoes is Halstead Metrics.

Coupling produced faults in system were very complicated to destroy than normal faults. Coupling is the measure of the degree of interdependence between two modules. Faults from one module to another module propagate faster if the coupling between two modules is more, so it decreases on the whole the system reliability. Reusability helps to possesses less coupling for particular module.

Cohesion measures the strength of a module. Cohesion demonstrates the interconnection between the elements of a module. Result of cohesion is low or high. Cohesion should be high makes software reliability. Reusability is the best remedy used in the module which leads highly cohesive. Highly cohesive modules are very easy to maintain. High cohesion component values makes fewer faulty and extremely reliable.

The complexity of the program is measured by Cyclomatic Complexity. System reliability is very poor because of Cyclomatic Complexity. Complexity is measured by developing a control flow graph and determining its linearly independent path, where nodes represent the number of statements in a program, and an edge represents the flow of data from one relation to another. The less complex system is easy to maintain and reuse.

A.Enhanced Adaptive Neuro Fuzzy Inference System (EANFIS)

The given component with defective profile can be estimated with the help of Enhanced Adaptive Neuro Fuzzy Inference System. Fuzzy Inference System has a top level of reasoning ability, which is made more robust using the low-level computational power of a neural network. Hybrid approach named EANFIS is developed by combining Fuzzy Inference System.

The proposed methodology is planned to build up a system which estimate the reliability for a specified set of inputs. Machine Learning Approaches are used to develop an intelligent system that repeatedly guess the system behavior based on a set of rules. Generate the rule based on observation of the previous data. Select the set of input parameters that damage the system. The rules are intended by the industry experts. The output of Machine Learning approaches depends on the rule. The structure of proposed methods contains the important steps.

Step 1: Input values are the selection of design metrics.

Step 2: Concern membership function is used to convert the design metrics into fuzzy inputs.

Step 3: Defining the rules based on expert ideas.

Step 4: Designing a fuzzy inference system based on rules.

Step 5: Develop the Enhanced Adaptive Neuro-Fuzzy Inference System (EANFIS)

Step 6: Find the faulty report of the components.

Step 7: Calculate the probability of the components.

Step 8: Check the preferred function in software doesn’t produce an occurrence of a fault.

Step 9: Finally it reflects the software reliability.

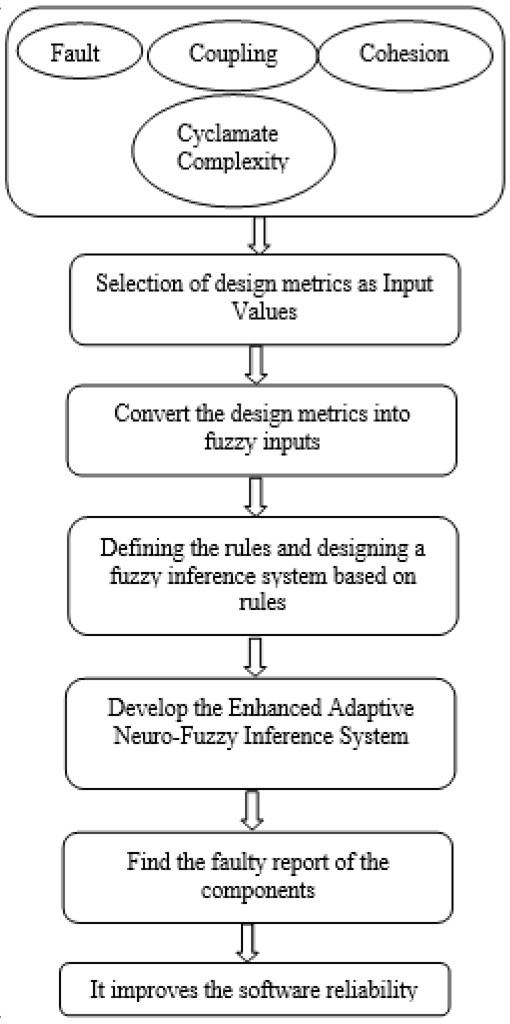
Fault
Coupling
Cohesion
Cyclamate Complexity
Selection of design metrics as Input Values
Convert the design metrics into fuzzy inputs
Defining the rules and designing a fuzzy inference system based on rules
Develop the Enhanced Adaptive Neuro-Fuzzy Inference System
Find the faulty report of the components
It improves the software reliability

Fig 1 : Architecture of Enhanced Adaptive Neuro-Fuzzy Inference System

B. EXPERIMENTAL DESIGN

The fuzzy logic contains a multi-defined logic which finds the intermediate values between true and false. It’s range from very low, low, medium, high, very high depending on the scale considered. A knowledge base is formed using the number of fuzzy rules (IF-THEN) in a fuzzy reasoning system. Inputs were collected as linguistic variables. The rules are formulated based on expert opinion and to control the divergence between the members. Majority voting approach was considered. The rule set decides the decision about whether a system is reliable or not. The proposed model concentrates four design metrics such as Faults, Coupling, Cohesion and Cyclomatic complexity as inputs and provides better value of reliability using the rule set. All inputs are classified as fuzzy sets low, medium and high respectively. Output reliability are classified as very low, low, medium, high, very high. All combinations of input are considered to form the rule set. Some of the rules are given below:

Rule 1: If Faults, the range between 0 to 2 is considered as Low, the range between 2.1 to 5 is considered as Medium, above 5 considered as High.

Rule 2: If Coupling, the range between 0 to 30 is considered as Low, the range between 30.1 to 60 is considered as Medium, above 60 considered as High.

Rule 3: If Cohesion, the range between 0 to 30 is considered as Low, the range between 30.1 to 60 is considered as Medium, above 60 considered as High.

Rule 4: If Cyclomatic Complexity, the range between 0 to 50 is considered as Low, the range between 50.1 to 100 is considered as Medium, above 100 considered as High.

Table 1: Fuzzy profile- Range of design metrics

Value	**Faults**	**Coupling**	**Cohesion**	**Cyclomatic Complexity**
LOW	0,0,2	0,24,30	0,22,30	0,33,44
MEDIUM	2,4,6	40.5,55,56	35,43,59	53,80,90
HIGH	6,12,18	67,70,82	64,73,79	105,110,130

Fuzzy Inference System structure is generated using the genfis2 function with the help of MATLAB Fuzzy Logic Toolbox wherein the input constituted four design metrics namely Faults, Coupling, Cohesion and Cyclomatic complexity and is also used to observe the output reliability for a particular set of input.

Fuzzy Inference System is educated using Artificial Neural Network and the output collected from Enhanced Adaptive Neuro-Fuzzy Inference System.

V. RESULTS AND DISCUSSION

The software metrics were calculated using the specified formulas. Assessment of these metrics can be done before the testing phase. The Fault Profile (FP) of a component is the output obtained from Enhanced Adaptive Neuro-Fuzzy Inference System. All components are not used in the software with the same frequency. The reliability of the software system will fall significantly if components with high fault profile are used frequently. The input is collected as linguistic variables and they are represented using the scale of 0 to 1. The 0.1 indicates least frequently used, 0.3 shows less usage, 0.5 points to medium usage, 0.7 represents high usage and 0.9 represents very high usage of the component. The probability of happening of fault is a result of Faulty Profile and Operational Profile. The probability of components that executes without failure is evaluated. The fault-free execution of component is directly proportional to the reliability of the software. The proposed EANFIS system predicts more components with highly reliable and fewer components with least reliable. Hence, accurate testing is required for components with less reliability. Estimation of reliability for each element is necessary.

VI. CONCLUSION

Reliability is the most essential feature that determines the ability of a particular critical software system to perform failure free operation. The major advantage of using fuzzy logic in modeling reliability

evaluating model is that it make use of expert knowledge in that domain based on which various if–then rules are being
designed. Enhanced Adaptive Neuro-Fuzzy Inference System (EANFIS) can be rightfully used for estimating the reliability. It improves the quality of the software. The software designer should recognize the metrics that distress the reliability of the software. The components are predicted at the premature stage that means before testing of the software. The output helped to reduce the testing attempt, reduce the cost and schedule the software for growth and maintenance.

REFERENCES

[1] Kanmani S, Uthariaraj V R, Sankaranarayanan V, Thambidurai P, "Object-oriented software fault prediction using neural networks", *Information and software technology*, Vol. 49, No.5, pp. 483-492, 2007.

[2] Jiang Y, Cuki B, Menzies T, Bartlow N, "Comparing design and code metrics for software quality prediction", *Proceedings of the 4th international workshop on Predictor models in software engineering*, pp. 11-18, ACM, May 2008.

[3] Tyagi K, Sharma A, "An adaptive neuro fuzzy model for estimating the reliability of component-based software systems", In *applied Computing and informatics*, Vol. 10, No.1, pp. 38-51, 2014.

[4] Sehgal R, Mehrotra D, "Predicting Faults before Testing Phase using Halstead's Metrics", *In development*, Vol.*9, No.7*, 2015.

[5] Jiang Y, Cukic B ,Menzies T, "Fault prediction using early lifecycle data", *In Software Reliability, ISSRE'07 the 18th IEEE International Symposium*, pp. 237-246, November 2007.

[6] Mahdie khadame, Sima Emadi, "Measurement of Software Reliability Based on Coupling and Cohesion Rate by External and Internal View of Classes", *International Journal of Computer Science and Network Security*, Vol.16 No.11,pp.7-15, November 2016.

[7] Zakarya Abdullah Alzamil, "Software Coupling and Cohesion Model for Measuring the Quality of Software Components", *Research Gate*,http://dx.doi.org/10.32604/cmc.2023.042711, Vol, 77, No.3, pp.3139-3161, December 2023.

[8] Kaur P, and Kaushal S, "A fuzzy approach for estimating quality of aspect oriented systems," *International Journal of Parallel Programming*, Vol.48, No. 5, pp. 850–869, 2020.

[9] Rizwan M, Nadeem A and Sindhu M, "Empirical evaluation of coupling metrics in software fault prediction", in *IEEE 17th Int. Bhurban Conf. on Applied Science and Technology*, Islamabad, Pakistan, pp. 434–440, 2020.

[10] Chowdhury I, Zulkernine M,"Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities", *Journal of Systems Architecture*, Vol.57, No.3, pp. 294-313, 2011.

[11] Surendra Kalagara, "Cyclomatic Complexity in Software Development", *International Journal of Engineering Research & Technology*, Vol.8, No.16, pp. 46-47, 2020.

[12] Ayman Hussein Odeh, Munther Odeh, Hussein Odeh,Nada Odeh,"Measuring Cyclomatic Complexity of Source Code Using Machine Learning", *Research Gate*, Vol.38, No.1, pp. 183-191, February 2024.