International Journal of Computer Applications (0975 – 8887)
Volume 127 – No.7, October 2015

Software Reliability Estimation of Component based Software System using Fuzzy Logic

Gopal Prasad Jaiswal
(Research Scholar)
Department of Computer Science and Engineering
RITEE Raipur, Chhattisgarh
CSVTU Bhilai, Chhattisgarh, India

Ram Nivas Giri
(Asst. Professor)
Department of Computer Science and Engineering
RITEE Raipur, Chhattisgarh
CSVTU Bhilai, Chhattisgarh, India

ABSTRACT
Software Reliability Modeling has been one of the much-attracted research domains in Software Reliability Engineering. Software reliability means provide reusable, less complex software, to perform a set of successful operation and his function within a provided time and environment. Software designers are motivated to develop reliable, reusable and useful software. In past, Object-Oriented Programming System (OOPS) concept is to be used in purpose of reusability but they are not providing powerful to cope with the successive changing as per requirements of ongoing applications. After that Component Based Software system (CBSS) is in floor. IT is based on reusability of his component with less complexity. This paper presents a new approach to analyze the reusability, dependency, and operation profile as well as application complexity of component-based software system. Here, we apply Fuzzy Logic approach to estimate the reliability of component-based software system with the basis of reliability factor.

Keywords
Component, Object-Oriented Programming System (OOPS), Component Based Software system (CBSS), Fuzzy Logic, Fuzzy Inference System (FIS), Adaptive Neuro Fuzzy Inference System (ANFIS), Reliability, Application Complexity, Component Dependency, Operation Profile, Reusability, Fuzzification, Defuzzification, Reliability Model, Rule Based Model, Path Based Model, Additive Model, etc.

1. INTRODUCTION
Software reliability is defined as the probability of failure – free software operation for a specified period of time in a specified environment. The reliability of a software product is usually defined to be "the probability of execution without failure for some specified interval of natural units or time" [1]. Software reliability is a feather of any software. Software reliability is depends on performance of successful operations and function as well as less complexity, maintainability, portability, flexibility and so on. Basically we can say that software reliability is a feather of the software that to be depend on another feather of the software. Hence, we cannot simply define it. In a binary form we can say that if software is correct and failure-free then its reliability is 1 else 0. Reliability is still predict probabilistically as

Software Reliability = [1-probability of failure]

Software reliability is mostly depending on reusability of the software because reliability of software is directly proportional to its reusability. For this purpose many year ago object-oriented programming system (OOPS) concept is appear for software development. But he was not successful as per requirement. After that another concept is appear in development floor that is Component Based Software System (CBSS).

Component Based Software System (CBSS) is a paradigm that aims at constructing and designing systems using a pre-defined set of software components explicitly created for reuse. Component based software development is most promising approach for software development today. This approach is based on the idea that software systems can be developed by selecting appropriate off-the-shelf components and then assembling them with well-defined software architecture [2]. This new software development approach is very different from the traditional approach in which software systems can only be implemented from scratch.

This paper presents soft computing techniques for reliability estimation for the component based software system. Here we will use fuzzy logic for estimating the reliability of the software. Fuzzy logic provides logical capabilities as well as learning capabilities for decision making. Logically decision that is Fuzzy Inference System (FIS) based on fuzzy rule and learning capability based on training for decision making that is Adaptive Neuro Fuzzy Inference System (ANFIS). In this paper we will use both type of facilities are adopted with different number of membership function for estimation component based software system and analysis that which one is provide better reliability for both the models.

Rest of the paper is sorted out as follows Region-2 related research work Region-3 proposed framework. Region-4 proposed methodology for CBS reliability. In Region-5, experiments, observation and result analysis of different approaches has been calculated. Paper is concluded with a summary and the description for future work in Region-6.

2. RELATED RESEARCH
In this software reliability estimation many number of models proposed for estimating CBSS reliability. We can summaries these approaches into three types [3]:

- Architecture Based Reliability Models
- Mathematical Model for Estimating CBSS Reliability
- Soft Computing techniques for estimating CBSS reliability

Architecture Based Reliability Models: Shooman, 1976 "Structural models for software reliability prediction", here consider the possible execution paths for estimating the reliability of an application. A sequence of components along different paths is obtained by either algorithmic or experimental testing [4]. Cheung, 1980 "A user oriented software reliability model" user-oriented software reliability figure of merit is defined to measure the reliability of a software system with respect to a user environment. The reliability of a system is expressed as a function of the reliabilities of its components and the user profile, Means that the current behavior of a component is independent of its previous behavior. These models consider transfer among components to be Markov behavior, which means that the current behavior of a component is independent of its previous behavior. These models can be represented in two ways, namely, as composite models or as hierarchical models [5]. Popostojanova and Trivedi, 2001; Cai et al., 2003; Gokhle, 2007 "Architecture based approach to reliability assessment of software systems" architecture-based reliability models such as state-based and path-based models and find out CBSS reliability depends not only on the architecture but also on the operational profile for the input[6] . Yacoub, S., Cukic, B., and Ammar, H., "Scenario based reliability analysis approach for component based systems" in 2004 propose an approach to reliability analysis called scenario based reliability analysis. This approach introduces component dependency graphs (CDGs) which can be extended for complex distributed systems. This approach is based on scenarios which can be captured with sequence diagrams, which means that the approach can be automated [7].

Mathematical Model for Estimating CBSS Reliability: Dong, W., Huang, N., Ming, Y., 2008 “Reliability analysis of component-based software based on relationships of components” a new model for estimating CBSS reliability in which various complex component relationships are analyzed. The Markov model is used to solve these complicated relationships, which have a large impact on a system’s reliability. The results were used to develop a new tool to calculate software application reliability [8]. Huang, N., Wang, D., Jia, X., 2008 “An algebra-based reliability prediction approach for composite web services” proposed a technique based on algebra which provides a framework for describing the syntax and predicting the reliability of a CBSS. If operational profiles have been changed, the loop times of iteration will be changed [9]. Goswami V., Acharya, Y.B., 2009 “Method for reliability estimation of COTS components based software systems” proposed an approach to CBSS reliability analysis which takes the component usage ratio, which is the time allotted for a component’s execution out of the application’s overall execution time, into consideration. This approach can be used in real-time applications [10]. Seth, K., Sharma, A., Seth, A., 2010 “Minimum spanning tree-based approach for reliability estimation of COTS based software applications” an algebra-based reliability prediction approach (Huang, N., Wang, D., Jia, X., 2008.) is to be used [11].

Soft Computing techniques for estimating CBSS reliability: Dimov, Aleksandar, Sasikumar, and Punnekkat, “Fuzzy reliability model for component-based software systems” in 2010 a fuzzy reliability model for Component Based Software System (CBSSs), based on fuzzy logic and probability theory. A mathematical fuzzy logic model was based on necessity and possibility is proposed to predict the reliability of a CBSS. This model does not require component failure data because it is based on uncertainty. However, a mechanism is necessary to model the propagation of failure between components and failure behavior [12]. Lo, J., 2010 “Early software reliability prediction based on support vector machines with genetic algorithms” proposed a software reliability estimation model based on an SVM and a GA. This model specifies that recent failure data alone are sufficient for estimating software reliability. Reliability estimation area for the SVM is determined by the GA. This model is less dependent on failure data than are other models [13]. Hsu, C., Huang, C., 2011 “An adaptive reliability analysis using path testing for complex component based software systems” proposed an adaptive approach for testing path reliability estimation for complex CBSSs. Path reliability estimation: these use sequence, branch, and loop structures. The proposed path reliability can be used to estimate the reliability of the overall application [14]. Tyagi, K., Sharma, A., 2012 “A rule-based approach for estimating the reliability of component-based systems” proposed an approach based on fuzzy logic for estimating CBSS reliability. In this approach, four critical factors were identified for estimating the reliability of a CBSS. They are used to design an FIS for the estimation [15]. Kirti Tyagi, Arun Sharma 2014, “An adaptive neuro fuzzy model for estimating the reliability of component-based software systems” propose a model for estimating CBSS reliability, known as soft computing model or an adaptive neuro fuzzy inference system (ANFIS), that is based on these two basic elements FIS and ANFIS, Here, we analysis its performance with that of a plain FIS (fuzzy inference system) for different data sets. This is a hybrid method that requires less computational time than traditional approaches and the previously proposed FIS approach. [3]

3. PROPOSED FRAMEWORK

In region-2 research work to read various models that to be proposed reliability estimation model and conclude that all the models have their own restriction to estimate the reliability of the Component Based Software System (CBSS). We have proposed an soft computing model But still soft computing model have various techniques are available. Some soft computing techniques are listed below:

- Fuzzy Inference System (FIS)
- Artificial Neural networks (NN) and Adaptive Neuro Fuzzy Inference System (ANFIS)
- Support Vector Machines (SVM)
- Probabilistic Reasoning (PR) or Probabilistic Logic (PL)
- Evolutionary Computation (EC)
- Evolutionary Algorithms (EA)
- K-Nearest Neighbor (K-NN)
- Genetic Algorithms (GA)
- Chaos Theory (CT)
- Hybrid Model

Our proposed soft computing model is based on fuzzy logic that to be overcome previously researched restriction and estimates the nearest reliability of the Component Based Software System (CBSS).

We are using fuzzy logic for software reliability estimation. Fuzzy logic is basically if-then rules syntactically. They will provide logical capabilities as well as learning capabilities for decision making. Logical decision that is Fuzzy Inference System (FIS) and learning capability based decision making that is Adaptive Neuro Fuzzy Inference System (ANFIS). In this paper we will use both type of facility for estimation component based software system. Here, we will explain both the soft computing technique one by one:

Fuzzy Inference System: A Fuzzy Inference System (FIS) is a way of mapping an input space to an output space using fuzzy logic. FIS framework is displayed at fig. 1. FIS uses a collection of fuzzy membership functions and rules, instead of binary logic, to reason about data. The rules in FIS (sometimes may be called as fuzzy expert system) are fuzzy production rules of the form [25] [26]:

if M then N, where M and N are fuzzy statements.

For example, in a fuzzy rule

if A is low and B is high then C is medium.

Here A is low; B is high; C is medium are fuzzy statements; X and Y are input variables; Z is an output variable, low, high, and medium are fuzzy sets.

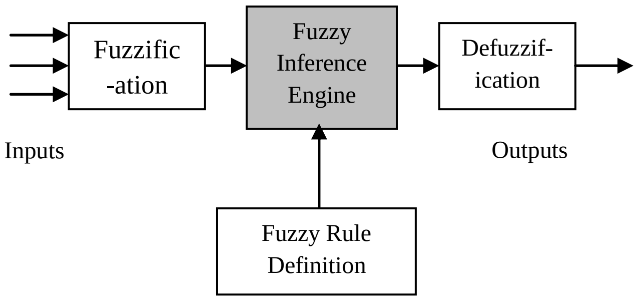

Fig. 1 Framework of Fuzzy Inference System

Adaptive Neuro Fuzzy Inference System: An adaptive neuro-fuzzy inference system or adaptive network-based fuzzy inference system (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. It was developed in the early 1990s [16] [17]. Since it integrates both neural networks and fuzzy logic rules, it has potential to grab the benefits of both in a single paradigm. This inference system is a set of fuzzy IF–THEN rules that have learning capability to approximate nonlinear functions[18]. Hence, ANFIS is considered to be a universal estimator [19]. Below figure-2 is basic ANFIS structure for two input variable with two membership function for each input variable [25].

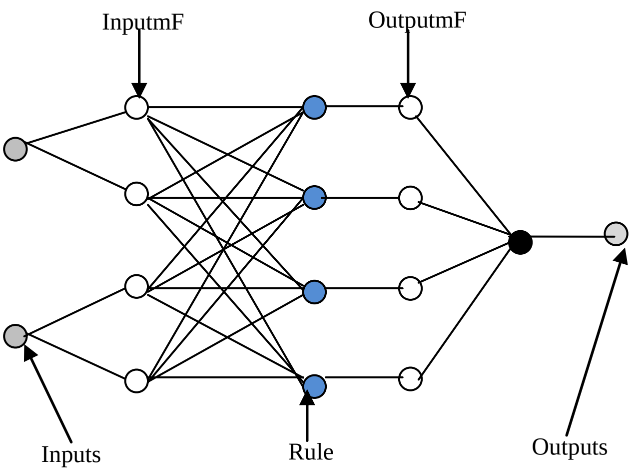

Fig. 2 Structure of Adaptive Neuro Fuzzy Inference System

4. PROPOSED METHODOLOGY

In this paper we will use soft computing techniques for software reliability estimation of Component Based Software System (CBSS). It paper is based on fuzzy logic based computing technique, and we are use FIS and ANFIS. This both the model is to performed in to some input variables. There so we will use some software feather for the calculation of the software reliability. Those feathers are listed below:

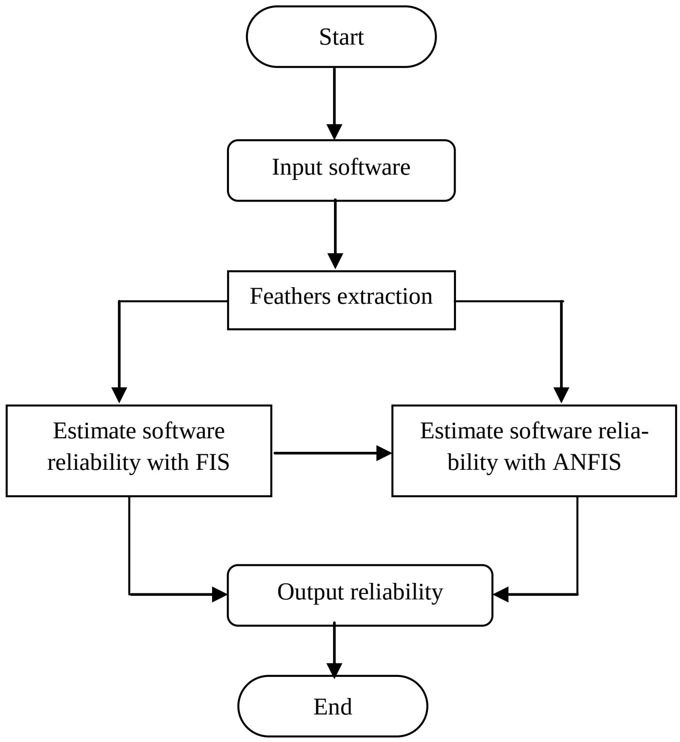

Fig. 3 Flow chart of proposed methodology

Reusability: Reusability means how to use any component in multiple times without any failure or any other restriction called software reusability. The reliability of a component is directly proportional to its reusability. Component reusability is calculated on the basis of components feathers [3] [20] [21] [22] [23] [24].

Component Reliability ∝ Reusability

Reusability of the any software will be based on attributes, sub-attributes and there selected metrics. Here we are discussed about reusability attributes or Evolutionary model [20] that is reusability of the software is depending upon various attributes. This attributes are listed below:

- Understandability
- Portability
- Maintainability
- Variability
- Flexibility

According to software Evolutionary mode,

Reusability of Package = [0.2*Understandability + 0.2*Portability + 0.2*Maintainability + 0.2*Variability + 0.2*Flexibility]

According to Reusability attribute model reusability of any package is calculating as follows:

Operation Profile: Operation profile means how much number of operations was performed successfully. It will be directly proportional to its reliability [3] [15].

Component Reliability ∝ Operation Profile

Component Dependency: Component dependency is feather of software. It gives information about how much component is dependent on another component [3] [15].

Table-I Software reliability analysis of FIS and ANFIS					
Inputs feathers				Output Reliability	
Application Complexity	Operation Profile	Component Dependency	Reusability	FIS model	ANFIS model
0.50837246	0.075433943	0.730419813	0.625417729	0.330568529	0.33056724
0.584072936	0.107107929	0.761829235	0.68519606	0.32900759	0.323148477
0.67895207	0.143440214	0.825335807	0.756732372	0.327833969	0.327830848
0.703270433	0.138480278	0.848301698	0.775563004	0.327794023	0.338613467
0.703988662	0.140793109	0.851358641	0.775565915	0.327908851	0.327011649

Component dependency $\propto$ (1 / reliability)

Application Complexity: Application complexity is feather of any software that gives information about complexity of the software. Application complexity is directly proportional to number of component [3] [15].

Application Complexity $\propto$ (1 / reliability)

After the calculating these above software feathers, we are applying FIS and ANFIS fuzzy soft computing technique in these calculated feathers (ex.-reusability, operation profile, and component dependency and application complexity) for reliability estimation of the Component Based Software System (CBSE). Figure-3 described flow chart of our proposed model that to be given in above Fig. 3.

5. EXPERIMENTS, OBSERVATIONS AND RESULT ANALYSIS

In this part, we are applying our methodology in between number of freeware software. We collected software data from www.sourceforge.net . Here we will use software data as a Jasmin and pBeans. Both the software are various versions are available in the www.sourceforge.net .After collecting the software data sets we are calculate the above described feather (ex.-reusability, operation profile, and component dependency and application complexity) for the estimation of software reliability.

After this we are applying our model that is FIS and ANFIS:

Fuzzy Inference System model: we are using describes feathers as a input data set and calculated software reliability with three and five membership function separately. In FIS with three membership function total 81rules defined for fuzzy inference engine and calculate software reliability. Similarly for five membership functions total 625 rules are defined for fuzzy inference engine and calculate software reliability with basis of three membership function and five membership function separately.

Adaptive Neuro Fuzzy Inference System model: we are using describes feathers as a input data set and give the respective output data or target for learning capability because ANFIS is supervised learner. ANFIS is applied for software reliability with three and five membership function separately.

In FIS with three membership function total 81rules are generated automatically for learning capability of inference engine, after that give the software reliability as per input data. Similarly for five membership function total 625 rules are generated automatically for learning capability of inference engine, after that give the software reliability as per input software data.

The software reliability analysis of FIS and ANFIS is to be listed in above Table-I.

6. CONCLUSION AND FUTURE SCOPE

We are estimate the reliability of component based software system (CBSS). CBSS reliability is to be estimated by the FIS and ANFIS with two different number of membership function. After compression of the output reliability values for different input sets, than we are analysis that FIS and ANFIS model is provide better result for five membership function as compare three membership function. Here, CBSS reliability estimation performed based on only four factors that is Reusability, Operational profile, Component dependency and Application complexity. But CBSS reliability affected by more other factor like Fault density, Software quality, Together with functionality, Usability, Availability, Performance, Serviceability, Capability, Install ability and Maintainability. So the addition of this factor is left for future work.

7. REFERENCES

[1] Musa, John. Software Reliability Engineering, New York, NY, McGraw-Hill, 1998.

[2] G. Pour, “Component-Based Software Development Approach: New Opportunities and Challenges,” Proceedings Technology of Object-Oriented Languages, 1998. TOOLS 26, pp. 375-383.

[3] Tyagi, K., Sharma, A., 2014, “An adaptive neuro fuzzy model for estimating the reliability of component-based software system”, applied Computing and informatics 10, 38–51.

[4] Shooman, M., 1976. Structural models for software reliability prediction, proceeding of the Second International Conference on Software Engineering, San Francisco, CA, pp. 268–280.

[5] Cheung, R.C., 1980. A user oriented software reliability model. IEEE Trans. Softw. Eng. 6 (2), 118–125.

[6] Popostojanova, K.G., Trivedi, K.S., 2001. Architecture based approach to reliability assessment of software systems. Perform. Eval. J. 45 (2), 179–204.

[7] Yacoub, S., Cukic, B., Ammar, H., 2004. Scenario based reliability analysis approach for component based systems. IEEE Trans. Reliab. 53 (4), 465–480.

[8] Dong, W., Huang, N., Ming, Y., 2008. Reliability analysis of component-based software based on relationships

of components, IEEE Conference on Web Services, pp. 814–815.

[9] Huang, N., Wang, D., Jia, X., 2008. FAST ABSTRACT: an algebra-based reliability prediction approach for composite web services, 19th International Symposium on Software Reliability Engineering, pp. 285–286.

[10] Goswami V., Acharya, Y.B., 2009. Method for reliability estimation of COTS components based software systems, International Symposium on Software Reliability Engineering.

[11] Seth, K., Sharma, A., Seth, A., 2010. Minimum spanning tree-based approach for reliability estimation of COTSbased software applications. IUP J. Comput. Sci. 4 (4), 13–21.

[12] Dimov, Aleksandar, Sasikumar, Punnekkat, 2010. Fuzzy reliability model for component-based software systems, 36th EUROMICRO Conference on Software Engineering and Advanced Applications, pp. 39–46.

[13] Lo, J., 2010. Early software reliability prediction based on support vector machines with genetic algorithms, Fifth IEEE Conference on Industrial Electronics and Applications, pp. 2221–2226.

[14] Hsu, C., Huang, C., 2011. An adaptive reliability analysis using path testing for complex component based software systems. IEEE Trans. Reliab. 60 (1), 158–170.

[15] Tyagi, K., Sharma, A., 2012. A rule-based approach for estimating the reliability of component-based systems. Adv. Eng. Softw. 54, 24–29.

[16] Jang, Jyh-Shing R (1991). Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm (PDF). Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14–19 2. pp. 762–767.

[17] Jang, J.-S.R. (1993). "ANFIS: adaptive-network-based fuzzy inference system". IEEE Transactions on Systems, Man and Cybernetics 23 (3). doi:10.1109/21.256541.

[18] Abraham, A. (2005), "Adaptation of Fuzzy Inference System Using Neural Learning", in Nedjah, Nadia; de Macedo Mourelle, Luiza, Fuzzy Systems Engineering: Theory and Practice, Studies in Fuzziness and Soft Computing 181, Germany: Springer Verlag, pp. 53–83, doi:10.1007/11339366_3

[19] Jang, Sun, Mizutani (1997) – Neuro-Fuzzy and Soft Computing – Prentice Hall, pp 335–368, ISBN 0-13-261066-3

[20] Fazal-e-Amin, Ahmad Kamil Mahmood, Alan Oxley 2012, "An Evolutionary Study of Reusability in Open Source Software", International Conference on Computer & Information Science (ICCIS) 967-972.

[21] Fazal-E-Amin, et al., "A Review of Software Component Reusability Assessment Approaches," Research Journal of Information Technology, vol. 3, pp. 1-10,20 1 1.

[22] Fazal-E-Amin, et al., "Using open source components m software product lines - An exploratory study," in IEEE Conference on Open Systems (ICOS) 2011, 20 1 1, pp. 332-337. [6] Fazal-E-Amin. et aI., "A mixed method study to identify factors affecting software reusabil ity in reuse intensive development," in National Postgraduate Conference (NPC), 2011,20 1 1, pp. 1-6.

[23] Capretz, et aI., "COTS-based software product line development," International Journal of Web Information Systems, vol. 4, pp. 165 - 180, 2008.

[24] Ahmed, et al., "A Model of Open Source Software-Based Product Line Development," in Computer Software and Applications. 2008. COMPSAC '08. 32nd Annual IEEE International, 2008, pp. 12 15 - 1220.

[25] Mehdi Neshat, Ali Adeli, Azra masoumi and Mehdi sargolzae, 2011,"A Comparative Study on ANFIS and Fuzzy Expert System Models for Concrete Mix Design", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011 ISSN (Online): 1694-0814, 196-210.

[26] R.Chinnaiyan and Dr.S.Somasundaram,2008, "RELIABILITY ASSESSMENT OF COMPONENT BASED SOFTWARE SYSTEMS USING TEST SUITE - A REVIEW", Journal of Computer Applications, Vol – 1, No.4, Oct – Dec 2008, 34-37.