A PERSPECTIVE APPROACH OF SOFTWARE RELIABILITY MODELS AND TECHNIQUES

Chahat Sharma and Sanjay Kumar Dubey
Department of Computer Science and Engineering, ASET, Amity University, Noida (U.P.), India
E-Mail: chahat.s.03@gmail.com

ABSTRACT

Software Reliability holds an important place in maintaining software quality. The efficiency of any software depends on its reliable nature. The evaluation of reliability is a prime function of any software system. A widespread research has been done and various methodologies exist to predict the reliability of software systems. This paper extracts relevant methodologies from various journals, conferences and transactions. It is a perspective approach to analyze the widely used models and techniques which are used to measure software reliability. The paper is mainly divided into five sections: elucidate the evolution of various models and approaches of software reliability, illustrates the object oriented metrics used for estimation of software reliability, the review approach, the literature review and the review results along with certain merits and issues which form basis to bridge the gap between the current and the past research done on software reliability. It also discusses about the future work to stretch the breadth of the relevant literature in order to conduct more research on the extensively used reliability techniques in software industry.

Keywords: software reliability, model, technique, metrics, assessment.

INTRODUCTION

Software reliability is an essential and crucial factor to estimate software quality. Software reliability is the probability that software will not cause the failure of a system for a specified time under specified conditions [1]. ISO/IEC 25010:2011 product quality model defines reliability as the degree to which a system, component or product performs specified functions under specified conditions for a specified period of time [2]. Reliability is a key factor among the eight functional characteristics of quality model that contributes to the efficiency of the software system. The reliability characteristic is further composed of four related sub-characteristics which are maturity, availability, fault tolerance and availability. Just like hardware, the reliability of software can be measured and evaluated [3]. The evaluation of reliability is very important as it contributes to the economic success of any software system. There are various approaches that can be used to measure the reliability of the software system. The software reliability is inversely related to software complexity [4]. Hence it is required to analyse different reliability metrics under different factors that can affect reliability.

In the recent years various software reliability growth models have been proposed [5, 6, and 7]. In general there are two reliability models - deterministic model and probabilistic model. The deterministic model is one which studies the number of operands and operators in the program whereas probabilistic model represents fault removal and failure occurrences. The probabilistic models are further classified into different categories, such as failure rate, error seeding and Non Homogeneous Poisson Process (NHPP). Among these NHPP has been widely used because it describes the failure phenomenon [8, 9, 10 and 11]. However, it poses certain shortcomings because of its stochastic behaviour on software failure process. An alternative solution to this problem is the use of neural networks.

The use of neural network models has a significant advantage over the analytical models as it requires only failure history as input and no priori assumptions. An extensive research has been done in past to predict the software reliability using neural networks. The usefulness of neural networks lies in the fact that user is required to gather the concerned data and invoke algorithms for training the neural network. The domain metrics which were used in neural network were not sufficient to predict the object oriented faults. Hence, software metrics in object oriented paradigm were used to improve the accuracy and reliability of software.

A perspective approach is aimed to study and analyse various methodologies used for the evaluation and estimation of software reliability. The remaining part of the paper is summarized as follows: Section 2 illustrates the object oriented metrics used to provide accurate and reliable software. Section 3 describes the review methodology approach. Section 4 is an analysis of the literature structured to enlist the techniques for measuring the software reliability. Section 5 concludes and discusses the future work on software reliability.

OBJECT ORIENTED METRICS

The object oriented design metrics used for evaluating object oriented design and selecting an optimal design among alternatives with respect to reliability are as [12,13].

Data Encapsulation

The classes are treated as black boxes in which the operations are defined in the internals of the black box. The details internal to the class are hidden from the clients of the class. The clients can see only public methods of the class. The instance variables of a class can be manipulated only by the methods of class which is possible when all instance variables are private members of the class. The encapsulation of data members and complexity of the class can be measured by the number of instance variables which are private members of the class.

a) **Number of Encapsulated Variables (EV)**: The number of instance variables that are private members of the class. The higher the number of instance variables, the better is encapsulation.

b) **Number of Non-Encapsulated Variables (NEV)**: The number of instance variables that are public members of the class. The higher the number of instance variables, the worse is the encapsulation.

Number of Methods (NM)

The number of methods of a class denotes the complexity of class. It is indicated by the number of operations (methods) which a class can support. The more the number of methods in a class, the more is its complexity.

Lines of Code per Class (SLOC)

The number of lines of code in a method represents size of the class interface. The SLOC is used to indicate the number of source lines of code for methods of a class.

Coupling between Classes (CBC)

The relationship between pair of classes exists when methods of one class uses the instance variables of other class. CBC for a class is the number of classes coupled to that class.

Depth of Inheritance Tree (DIT)

The depth of inheritance of a class is the longest path in the graph which originates from the node representing the graph. The structure of inheritance of classes can be represented as acyclic directed graph, where the nodes represent classes. The DI indicates the dependence of the class on class hierarchy. The hierarchy of the depth is represented in the form of a tree.

Number of Subclasses (NSC)

The classes designed to be generic can have subclasses while a class that is very specialized do not contain any subclass. The reusability amount of a class is estimated by the number of derived classes in the class. The number of subclasses in most cases decreases as the depth of inheritance for class increases.

Lack of Cohesion of Methods (LCOM)

This metric shows the use of common instance variables in the methods of a class. The methods such as M_i and M_j of a class are cohesive in nature if they share one or more instance variables. The lack of cohesion can split the class into two or more classes containing the same functionality as the original class. Hence the LCOM denotes the cohesive of classes sharing common instance variables.

Number of Children (NOC)

It measures the number of post descendants of a particular class. This leads to measurement of the reusability of class. The more reusable a class, more is the complexity as many classes are affected due to the modifications after implementation.

Coupling Between Objects (CBO)

Coupling between Objects refers to the coupling of one class to other classes.

Weighted Methods per Class (WMC)

Weighted Methods per Class refers to the number of member functions and operators which are defined in every class.

REVIEW METHODOLOGY

Inclusion Approach

The papers based on software reliability are only included aiming to study, analyse and improve the software reliability process. The journal papers are included in our review to describe the research based on software reliability. The papers other than the improvement of the software reliability are not included. During the analysis certain papers based on similar approach and experiments were witnessed. Such papers are excluded to prevent ambiguity in the review.

Inclusion of Research Papers

The identification of research papers were based in the context of objective, issue, manual reading of titles, abstract and conclusion of all published papers in various journals, like IEEE, ACM, Springer and Elsevier etc. and in conferences. Some papers were identified through the reference list of relevant papers while some were extracted from the digital library of IEEE, ACM and Springer. Both the authors searched for good journals potentially. The papers were combined together to meet the objective of the research.

The papers which were more important for inclusion in our review were read in detail to determine its effectiveness with respect to the literature review. Out of 78 papers 41 papers were extracted from the relevant journals, transactions and conferences etc.

Categorization of Papers

In order to analyse, the research papers have been classified according to attributes as well as categories listed in Table-1. The categories are selected according to the requirement of analysis.

Table-1. Categorization of papers.

Attributes	Categories
Research Area	Models, Visual Representation, Examination, Approach, Statistical Estimations, Datasets, Others
Research Application	Survey, Experiments, Principles, Review, Evaluation, Case study
Participants	Employees, Students, Not Applicable

The tabular schema of classification was developed for the purpose of review. It is not meant to be a general purpose classification of reliability studies. The classification may provide substantial help to other researchers searching for related papers. Most of the categories are non-exclusive in nature i.e. a paper may take into consideration more than one reliability approach or apply more than one research model.

The classification of the reliability based papers is improved in the best possible manner while the descriptions of classification are subject to change further. However, the categorization of classification is considered as whole for the purpose of analysis.

ANALYSIS

In the work by Thwin and Quah [13] a study was conducted to predict the software development faults using object oriented metrics. The experiment was conducted on three industrial real systems. The data set taken was divided into training set, production and test-set. The computations of the patterns derived from the training set were investigated using Ward Network [14]. The neural network is aimed to identify the faults in object oriented metrics concerning with the inheritance related measures, complexity measures, coupling measures and memory allocation metrics defined by [15], [16]. The accuracy of the model was predicted using multiple regression models.

Aggarwal and Gupta [17] explained the concept of neural network approach to measure the reliability of software modules. The role of neural network based on the study of failure associated with the code and environment was discussed [18]. The techniques such as threshold acceptance based neural network, P-Sigma Network (PSN), Multivariate Adaptive Regression Splines (MARS) etc., were compared with the neural network technique. Due to performance problem in every technique ensemble models were developed to forecast the software reliability accurately.

Arora and Choudhary [19] used feed- forward neural networks to predict software reliability. The neural network was tailored using the static approach in the architecture. In the static approach, the architecture of network was defined in advance which remains invariant throughout the training phase. The training data by Yoshiro Tohma which described the prediction accuracy of neural network was used. The network was trained with 40 random seeds for given each training set size and their predictions were averaged. The percentage prediction error against execution time shown by feed forward network showed the relative prediction error.

Pandey and Ahlawat [20] presented the use of neural networks for predicting the software reliability and maintainability using object oriented metrics. The neural network architecture was used emphasizing on Back Propagation Network algorithm to find out possible errors at the output. The paper focused on the internal product metrics in comparison to the traditional metrics as it lacks certain important object oriented concepts such as data centric encapsulation, inheritance polymorphism [6]. The object oriented metrics such as size metrics, inheritance metrics, cohesion metrics and coupling metrics are used as the independent variables for the estimation and prediction reliability and maintainability.

Sharma and Bano [21] found software defects/ faults, requirement analyse (feasibility, survey methods, interview), cost, size estimation as the potential reliability factors affecting the reliability of a software. The source of data collection formed the basis of software reliability. The defect reports were used as the data which was collected in the form of survey questionnaire conducted in six different IT organizations.

In the review of the reliability measurement of object oriented design by Gupta and Kumar [22] software reliability facts were gathered to bridge the gap between the current researches to present framework for future examination. The software reliability evaluation is done listing the possible issues, function and knowledge required by the software engineer while executing a life cycle reliability management plan. The systematic review concludes to emphasize on reducing the effort in measuring reliability of object oriented design to deliver quality software in estimated time and budget.

In the research work by Hudli and Hudli [12] the software reliability of product is evaluated using software metrics. The complexity of object oriented software was measured by defining metrics for object-oriented design. The reliability of four software projects in the healthcare domain was calculated respectively. The results showed the complexity introduced by object oriented metrics, which should be considered in the reliability models for the software.

Kotiah and Khan [23] performed a survey on software reliability assessment using different machine learning techniques. Various machine learning techniques or approaches such as artificial neural network, fuzzy network, genetic algorithm, neuro-fuzzy approach, support vector machine (SVM), Bayesian classification, self-organizing map approach were used to compute the performance of machine learning techniques for prediction of software reliability. The results validate the effectiveness and efficiency of the machine learning approaches.

Schneidewind [24] comprehended the use of UML diagrams as the mathematical software for providing good view of design process. The UML diagrams do not lack comprehensibility and is an easier way to integrate all the relevant details of the software reliability design. The O-O paradigm was found to be less advantageous in comparison to other paradigm such as structured design for mathematical software.

Khatetneh and Mustafa [25] developed a new fuzzy expert system to predict software failures. The model was based on the growth of the reliability which focused on particular dataset behaviour in predicting software reliability. The results obtained from the experiment showed that the developed model predicted more accurate results of the target dataset in most points.

In the experimental work, Kiran and Ravi [26] assessed the software reliability based on ensemble models. One non-linear ensemble and three linear ensembles were designed and tested. Intelligent techniques such as Back Propagation trained Neural Network (BPNN), dynamic evolving neuro-fuzzy inference system (DEFNIS) and tree nets as well as statistical techniques such as Multivariate Adaptive Regression Splines (MARS) and Multiple Linear Regression (MLR) constituted the ensembles. The experiments based on software reliability from the literature showed that non-linear ensemble outperformed linear ensembles. Hence ensembles can be used as an alternative method to predict software reliability.

Antony and Dev [27] used CK metrics for measuring the reliability of software. Using the tool Java Class Analyzer the values of metric parameters from the source code were extracted to establish a relationship between reliability and object oriented metrics. The results from the assessment proved that by keeping low values of RFC, WMC, LCOM, CBO, DIT and high value of NOC software designers can achieve high reliability of the system. Hence CK metrics can be used as indicators of quality of the system.

In the systematic review by Singhal and Singhal [28] reliability is found to be quality factor that predicts the effort needed for software testing. The reduction in effort to measure reliability of object oriented design is highly important for delivery of reliable software within estimated budget and time. The results were concluded from the literature reviewed in the context of the reliability models and the frameworks used for measuring the reliability of OO design during early phases of software development lifecycle.

Dolbec and Shepard [29] gave a mathematical model which is a component based software reliability model. This model takes into consideration the execution paths in order to enhance the software reliability. The paper shows the simplification of previous work to structure software reliability on execution paths and converts it into component based model using component reliabilities and component usage ratios.

In the research done by Kumar and Dinker [30] object oriented features are described as class level metrics. Since the complexity of any software depends on the object oriented metrics a relation between reliability and CK metrics was established. The two strategies were proposed to correlate the factors of complexity of software under test and the effect of case-suit on the object oriented software reliability models.

Dadhich and Mathur [31] focused on the cross cutting factors in a distributed system to measure the reliability of the aspect oriented system. The implementation showed the improvement quality of software, higher productivity, cost savings and a better understanding through the use of fuzzy logic approach.

Alvarez and Aleman [32] discussed the use of software modelling approaches which manages the UML features before the development of UML for improving the software reliability. Nowadays various tools have been developed for modelling such as Rational Rose.

In the work by Khatri, Chhillar and Chhikara [4] object oriented system was taken under testing. The datasets included Squirrel SQL and SQL for Python on which the model worked. The results showed the improvement in reliability of software after testing on the bugs.

Nagar and Thankachan [33] discussed the reliability in the context of software medicine and manufacturing which covers the software structural quality and software functional quality. Some previous development models (Spiral development, prototyping, waterfall model) and advanced techniques (AGILE) are analyzed. The paper proposes algorithm for appropriate selection of the model for improving software reliability.

Gaudan, Motet and Auriol [34] showed the advancement in the object oriented metrics for software reliability. Some new metrics were presented to assess the complexity of object oriented models. The assessment of reliability was based on two approaches, statistical metrics and new MESS metrics which is a good predictor of fault risk.

Tyagi and Sharma [35] discussed the component based reliability estimation of system. Thirteen models such as Everett’s model, Kubat’s model, Chemg’s model, Gokhal’s model, Lettlewood’s model were discussed for estimation of reliability. To calculate the reliability of the component paper proposed two major points, firstly, reliability of individual component, secondly operation profiles of the system.

In the research work by P. Xu and S. Xu [36] a reliability model with object oriented paradigm was developed using mutation operation of Inheritance, Polymorphism, Dynamic Binding and Information Hiding. The complexity of software was calculated using method overloading. The metrics used for testing effectiveness and complexity were CBO, NOC, LCOM, WMC and DIT.

Singh, Khatri and Kapur [37] described the research in the field of object oriented approach for the closed source software. The work to develop reliability growth model under concurrent distributed development environment based on the open source software uses object oriented approach before development. It is a mathematical model that took use of the reported data of bugs of SQL for Squirrel and SQL for Python.

Mishra and Dubey [38] elucidated the methodology for evaluation of reliability of object oriented software system using fuzzy approach. The evaluation was done by taking ISO/IEC-9126 as the base model and object oriented metrics for the experiment. The reliability of object oriented software was evaluated by applying AHP and fuzzy layered approach.

Mishra and Dubey [39] in fuzzy qualitative evaluation of reliability of object oriented software system evaluated the reliability of object oriented software system. The CK metrics are considered and mapped with the sub-characteristics of reliability. The reliability of the object oriented software is evaluated using AHP approach.

Khoshgoftarr *et al.* [41] introduced the use of the neural network as a tool for predicting software quality. They found neural network model to show better predictive accuracy after comparing the neural network model with a non-parametric discriminant model. The domain metrics derived from the complexity metric data were used in their model. However, these metrics are not sufficient to predict the object oriented faults.

The comparison of the various methodologies, techniques and approaches to improve software reliability are shown in Table-2.

Table-2. Comparison of literature review.

S.N.	Year	Author (s)	Methodology	Dataset	Techniques	Merits	Demerits
1.	2002	M.M.T. Thwin and T.S. Quah [13]	Software development faults using object oriented metrics for reliability and maintainability prediction.	Programs of three subsystem of Human Machine Interface (HMI) software.	Multiple regression model and neural network model.	The object oriented metrics can be used to predict the number of development faults.	No issues
2.	2007	N.R. Kiran and V. Ravi [26]	Software reliability prediction based on ensemble models.	Software failure data	BPNN, DEFNIS, MARS and MLR	Non-linear ensemble models can be used as an effective alternative for predicting software reliability.	The hardware failures were not addressed.
3.	2009	K. Khatatneh and T. Mustafa [25]	Exploration of software reliability model.	Custom set of test data from command and control applications.	Fuzzy logic technique.	Accurate prediction of the software failures.	No issues
4.	2009	N.F. Schneidewind [24]	Use of UML diagrams for providing good view of design process.	Software failure data	Object oriented design approach	UML diagrams can be used to represent software reliability models.	OO- paradigm is complex for mathematical software
5.	2011	S.A. Hudli and A.V. Hudli [12]	To estimate reliability of software by addressing the complexity of object oriented design	Four software projects from healthcare domain	Metrics of object oriented design were defined.	Complexity of object oriented software should be considered to measure reliability of software.	Testing effectiveness should also be considered.
6.	2011	A. Singhal and A. Singhal [28]	Improvisation in software reliability research.	141 papers in 34 journals on software reliability.	Systematic review	Increased breadth of relevant studies, selected set of essential papers and more studies on commonly used technique in software industry.	Adherence to recommendations does not necessarily leads to better software reliability prediction.
7.	2011	Y. Wu and R. Yang [40]	Prediction of software reliability	Data set from a NASA supported project	General Regression Neural Network	More accurate and reasonable prediction model	No issues
8.	2012	B. Kotaiah and R.A. Khan [23]	Survey on software reliability assessment	Software failure datasets of projects	ANN, fuzzy network, genetic algorithm, SVM, Bayesian classification	Machine learning techniques can be used for software reliability model	Decision region, self-organizing map and neuro fuzzy approach needs to be used.
9.	2012	S. Khatri, R.S. Chhillar and A. Chhikara [4]	Prediction of reliability of object oriented system	Squirrel SQL and SQL for Python	Testing on bugs	Improvisation in the reliability of software	No issues
10.	2013	G. Aggarwal and V.K. Gupta [17]	To measure the reliability of software modules	Four different variants of ensembles	Neural network approach	Non-linear ensemble neural network outperformed other neural network	No model was discussed related to work.

11.	2013	M. Arora and S. Choudhary [19]	To predict the software reliability.	Dataset from real project by Yoshiro Tohma.	Feed forward neural network approach	Black box approach and the user need not know the underlying failure process of project.	Yet to be applied over wide range of software projects.
12.	2013	A. Pandey and A. Ahlawat [20]	To predict the software reliability and maintainability using object oriented metrics	Software product source code and design documents	Back Propagation Neural Network approach	Use of object oriented metrics to evaluate the quality of software.	No issues
13.	2013	N. Sharma and P. Bano [21]	Estimation of reliability factor issues of data with respect to different metrics of software process model.	Survey questionnaire results from six different IT organizations on the defect reports of software processes.	Analysis process	Design a reliable system using software technique.	No issues
14.	2013	J. Antony and H. Dev [27]	Measuring the reliability of software	Metric values taken from the projects.	Use of CK metrics for assessing the reliability of software	Keeping low values of RFC, WMC, LCOM, CBO, DIT and high value of NOC achieve in reliable software	Size of dataset is small and it did not validate metrics such as QMOOD and MOOD etc.
15.	2014	N. Gupta and R. Kumar [22]	Measurement of reliability in object oriented design	Reliability facts from current research.	Systematic review	Reducing the effort in measuring reliability of object oriented design delivers quality software	Failures in software can wear out and the product may not be robust on purpose.
16.	2014	A. Mishra and S.K. Dubey [38]	Evaluation of reliability of object oriented software using fuzzy approach.	ISO/IEC 9126 model and CK metrics	AHP along with fuzzy approach.	Evaluation of reliability of object oriented software systems by using CK metrics and ISO/IEC 9126 model	No issues

CONCLUSIONS

This paper reviews software reliability from various sources like journals, conferences, transactions and company specific journal and then classified according to research area, research application and study relation. On the basis of the analysis, software reliability is observed as an important quality factor in which object oriented metrics play an important role in software reliability prediction.

After substantial review it was found that object oriented metrics along with the neural network technique is effective for delivering quality software within estimated time and budget. In future, work will be carried out to assess the reliability of object oriented software system based on soft computing techniques to widen the breadth of research in software reliability.

REFERENCES

[1] ANSI/IEEE Std 729-1983. IEEE Standard Glossary of Software Engineering Terminology. The IEEE, Inc, New York.

[2] https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en, last accessed on-6 April 2015.

[3] M. Jedlicka, O. Moravcik and P. Schreiber. 2008. Survey to Software Reliability. Central European Conference on Information and Intelligent Systems, CECIIS.

[4] S. Khatri R.S., Chillar and A. Chhikara. 2012. Analyzing the Impact of Software Reliability Growth Models on Object Oriented Systems during Testing.

International Journal of Enterprise Computing and Business Systems. Vol. No. 2, Issue No. 1.

[5] A.L. Goel. 1985. Software Reliability Models: Assumptions, Limitations, and Applicability. IEEE Trans. Software Engineering.

[6] J.D. Musa, A. Ianino and K. Okumoto. 1987. Software Reliability - Measurement, Prediction, Applications. McGraw-Hill, Inc.

[7] Y.K. Malaiya and P.K. Srimani. 1990. Software Reliability Models: Theoretical Developments, Evaluation and Applications. IEEE Computer Society Press.

[8] A. L. Goel, and K. Okumoto. 1979. Time-Dependent error-Detection Rate Model for Software Reliability and Other Performance Measures. IEEE transactions on Reliability. Vol. R-28, No. 3, pp. 206-211.

[9] M. Ohba, et al. 1982. S-shaped Software Reliability Growth Curve: How Good Is It? COMPSAC'82, pp. 38-44.

[10] S. Yamada and S. Osaki. 1983. Reliability Growth Models for Hardware and Software Systems Based on Non-homogeneous Poisson Processes: a Survey. Microelectronics and Reliability. 23, pp. 91-112.

[11] S. Yamada and S. Osaki. 1983. S-shaped Software Reliability Growth Model with Four Types of Software Error Data. Int. J. Systems Science. 14, pp. 683-692.

[12] S.A. Hudli and A.V. Hudli. 2011. Impact of complexity of object-oriented design on software reliability," In proceeding of International Conference on Quality and Reliability Engineering ICQRE.

[13] M.M.T. Thwin and T.S. Quah. 2002. Application of Neural Network for Predicting Software Development Faults Using Object- Oriented Design Metrics. Proceedings of the 9th International Conference on Neural Information Processing (ICONIP). Vol. No. 5.

[14] L. Briand, J. Wiist, J.W. Daly and V. Porter. 2000. Exploring the Relationships between Design Measures and Software Quality in Object-Oriented Systems. Journal of Systems and Software.

[15] S.R. Chidamber and C.F. Kemerer. 1994. A Metrics Suite for Object Oriented Design. IEEE Transactions on Software Engineering. Vol. No. 20.

[16] M.H. Tang, M.H. Kao and M.H. Chen. 1999. An empirical study on object-oriented metrics. Proceedings of the Sixth IEEE International Symposium on Software Metrics.

[17] G. Aggarwal and V.K. Gupta. 2013. Neural Network Approach to Measure Reliability of Software Modules: A Review. International Journal of Advances in Engineering Sciences. 3(2).

[18] D.E. Eckhardth *et al. 1991.* An experimental evaluation of software redundancy as a strategy for improving reliability. IEEE transactions on software engineering.

[19] M. Arora and S. Choudhary. 2013. Software Reliability Prediction Using Neural Network. International Journal of Software and Web Sciences (IJSWS).

[20] A. Pandey and A. Ahlawat. 2013. Reliability and Maintainability of Software using Object Oriented Metrics by Artificial Neural Network. International Journal of Engineering and Management and Technology (IJEMT). 1(1).

[21] N. Sharma and P. Bano. 2013. A Survey of Software Reliability Factor. IOSR Journal of Computer Engineering. 12(1).

[22] N. Gupta and R. Kumar. 2014. Reliability Measurement of an Object Oriented Design: A Systematic Review. International Journal of Scientific Engineering and Technology. 3(12).

[23] B. Kotaiah and R.A. Khan. 2012. A Survey on Software Reliability Assessment by Using Different Machine Learning Techniques. International Journal of Scientific and Engineering Research (IJSER). 3(6).

[24] N.F. Schneidewind. 2009. Analysis of object-oriented software reliability model development. Innovation in Systems and Software Engineering. 5(4).

[25] K. Khatatneh and T. Mustafa. 2009. Software Reliability Modeling Using Soft Computing Technique. European Journal of Scientific Research. 26(1).

[26] N.R. Kiran and V. Ravi. 2007. Software reliability prediction by soft computing techniques. The Journal of Systems and Software.

[27] J. Antony and H. Dev. 2013. Estimating Reliability of Software System Using Object- Oriented. International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR).

[28] A. Singhal and A. Singhal. 2011. A Systematic Review of Software Reliability Studies. Software Engineering: An International Journal (SEIJ).

[29] J. Dolbec and T. Shepard. 1995. A component based software reliability model. In: Proc. of the 1995 Conference of the Centre for Advanced Studies on Collaborative Research (CASCON).

[30] D. Kumar and A.G. Dinker. 2014. Enhancement of Reliability in Object-Oriented Software Reliability Model. International Journal of Advanced Research in Computer Science and Software Engineering. 4(2).

[31] R. Dadhich and B. Mathur. 2012. Measuring Reliability of an Aspect Oriented Software Using Fuzzy Logic Approach. International Journal of Engineering and Advanced Technology. 1(5).

[32] A.T. Alvarez and J.L.F. Aleman. 2011. Improving System Reliability via Rigorous Software Modeling: The UML Case. In Proceedings of the 2001 IEEE Aerospace Conference, IEEE Computer Society.

[33] P. Nagar and B. Thankachan. 2011. Software Reliability Engineering–A Review. International Journal of Applied Physics and Mathematics. 1(2).

[34] S. Gaudan, G. Motet and G. Auriol. 2008. Metrics for Object-Oriented Software Reliability Assessment - Application to a Flight Manager. Proceedings of the Seventh European Dependable Computing Conference.

[35] K. Tyagi and A. Sharma. 2011. Reliability of Component Based Systems - A Critical Survey. ACM SIGSOFT Software Engineering Notes. 36(6).

[36] P. Xu and S. Xu. 2010. Reliability Model for Object-Oriented Software. 19th IEEE Asian Test Symposium.

[37] V.B. Singh, S. Khatri and P.K. Kapur. 2010. Reliability Growth Model for Object Oriented Software Developed under Concurrent Distributed Development Environment. 2nd International Conference on Reliability, Safety and Hazard (ICRESH).

[38] A. Mishra and S.K. Dubey. 2014. Evaluation of Reliability of Object Oriented Software System Using Fuzzy Approach. 5th International Conference-Confluence, The Next Generation Information Technology Summit (Confluence).

[39] A. Mishra and S.K. Dubey. 2014. Fuzzy Qualitative Evaluation of Reliability of Object Oriented Software System. IEEE International Conference on Advances in Engineering and Technology Research (ICAETR), IEEE.

[40] Y. Wu and R. Yang. 2011. Study of Software Reliability Prediction Based on GR Neural Network. 9th International Conference on Reliability, Maintainability and Safety (ICRMS), IEEE.

[41] P. M. Khoshgoftaar, E.B. Allen, J.P. Hudepohl and S.J. Aud. 1997. Application of neural networks to software quality modeling of a very large telecommunications system. IEEE Transactions on Neural Network. Vol. No. 8.