International Research Journal of Mathematics, Engineering and IT

ISSN: (2349-0322)

Association of Academic Researchers and Faculties (AARF)

Impact Factor- 8.132 Volume 12, Issue 10 Oct 2025

Website- www.aarf.asia, **Email** : editoraarf@gmail.com

A Critical Review of the Software Reliability Analysis Prediction

Dr. Naresh Kumar[1], Dr. Amrita Agrawal[2]

[1]Principal, Department of Mathematics, Goswami Ganesh Dutt Sanatan Dharam College, Palwal (Haryana)

[2]Assistant Professor, Department of Mathematics, Saraswati Mahila Mahavidyalaya, Palwal (Haryana)

Corresponding author: Dr. Amrita Agrawal[2]

DOI:DOI:18.A003.aarf.J14I01.010185

ABSTRACT: Software reliability is the ability of a system to carry out its necessary functions for a predetermined amount of time under static conditions. These days, software systems are used not only in important industries but even in small businesses because of features like quality determination and improvement. Reliability prediction, which offers quality improvement, resource allocation for development, and confidence to create high-quality software products, must be carried out in the early stages of the development process. The current situation has led to a gradual shift in the software development process to component-based development, which makes software more complex. A new development method utilizing reliability prediction models is needed to create software that is failure-free because the current development process is only appropriate for certain projects. This article's primary goal is to propose reliability prediction models for producing software that is error-free.

KEYWORDS: Software reliability, failure-free, system failures, producing software, reliability, availability, Prediction Models

1. INTRODUCTION

Reliability determines whether a software product is accepted or rejected. In order to reduce development costs and compete economically worldwide, practitioners strive to improve software reliability. Reliability prediction, which provides quality improvement, resource allocation for development, and testing objectives, is an early stage of the software development process. Reliability determines the quality of software. To put it another way, software product acceptance benefits from reliability prediction. It gauges how well software

can fix errors. However, some mistakes made throughout the software development process cannot be undone. Such elements that can detect errors early in the development process should be incorporated by developers. Additionally, errors are categorized according to their criticality and attempts are made to reduce them in the management process. Such a management strategy can be used to anticipate software reliability and reduce mistakes and malfunctions. Reliability models and their function in real-time systems are the main topics of this essay. Software reliability prediction models, which offer a wealth of information to assess reliability, can be used by developers. Numerous writers have created growth models for software reliability. However, not all growth models can be implemented, with the exception of certain projects or data sets. As a result, a software reliability prediction model that works with all data sets is being sought after. Examining the suitability of each software reliability prediction model that incorporates characteristics like dependability in relation to problems and suggestions for improvement is the primary responsibility. The reliability analysis models that are currently in use are the exclusive topic of this research. This can be used in the process of developing software in real time. A number of factors, including attributes, design complexity, reuse, code complexity, post-delivery faults, inspection, execution operation, and process and product metrics, are used in the creation of all dependability models. Software dependability models come in two primary varieties: probabilistic and deterministic. The number of unique operators and operands, faults, and machine instructions in a program are all examined using the deterministic model. Deterministic performance metrics are derived from program texture analysis and do not depend on random events.

2. SOFTWARE RELIABILITY: MODELLING

In present days, the utilization of expansive systems, for example, vitality circulation systems, media transmission systems, computerized PC Systems, space tests and multi-functioning estimating types of gear which required high level of exactness are expanding step by step. Further, the right working of these systems over a more drawn out interim of life is ending up increasingly more imperative as individuals are winding up progressively reliant on such systems. Other than this, our specialized systems are put to use in threatening conditions for instance applications in procedure industry (heat, humidity and so on) and furthermore there are assortments of circumstances where no upkeep is possible as a result of a separated area, for example, unmanned cold climate stations, remote space tests, and submerged enhancement stations in transatlantic links. Practical world class execution starts with a strong establishment that incorporates dependable resources, steady & repeatable business & work forms, and a

decidedly ready, connected with work constrain. Unwavering quality is a wide term that center's around the size of a thing to play out a required capacity under expressed condition for an expressed time of time (BS 4778, 2012). Scientifically, expecting that an entity is playing out its intended function at period rises to zero, reliability can be pigeonholed as the probability that a thing motivation keep on playing out its expected capacity without failure for a predetermined time border under articulated conditions. A system characterized here could be an electronic or mechanical equipment item, a programming item, an assembling procedure, or even an administration. The meaning of reliability has stood discussed much in fiction. In universal, two distinct meaning of reliability can stay given, concentrating each on the maker's point of view, or on the client's viewpoint. As of the producer's outlook, reliability is characterized as a normal for a thing (among others Birolini (2007), Lewis (1996)) i.e.: "the probability that a framework will accomplish its planned capacity for a specified historical of spell under an agreed arrangement of conditions". Verifiable this definition is presumption that lack of quality is brought about by an item disappointment, & that item disappointment is an unambiguous thought, since required capacity, conditions & time interim are unequivocally referenced& characterized preceding use. In this thesis, the consideration on dependability lays on unwavering quality a typical for the particular thing age.

Software reliability refers to the software's capacity to function without errors in a certain environment for a predetermined amount of time. Furthermore, the software consistently performs as predicted in the set criteria. Software quality is determined via reliability assessment. Every firm uses software systems extensively. Reliability is necessary to increase the productivity of such a system. Reliability models include insights about how to identify and eliminate errors as well as environmental factors that reduce failure rates. While software dependability steadily increases throughout the development process, hardware reliability is difficult to anticipate over time. Software design, inadequate quality control, the market, lucrative targets, and engineering design estimation are examples of software failures. A crucial component of software quality is software reliability. Three components make up the study of software reliability: modeling, measurement, and improvement. Software dependability modeling has advanced to the point where applying appropriate models to the problem can yield significant results. Since software dependability cannot be tested directly, software reliability is estimated and compared between products using other related characteristics. Improving software reliability is challenging. The amount of work that can be done to increase software reliability is severely limited by practical time and financial restrictions. There are two categories into which all dependability modeling falls. The many stages of modern

software development techniques are represented by black box models. All of the requirements for component-based applications cannot be met by this. White box models might meet every requirement for component-based software. It can now be used across the software development life cycle.

3. TYPES OF THE SOFTWARE RELIABILITY PREDICTION

Software reliability analysis prediction types include model-based, which use historical data or metrics to forecast reliability, and metric-based, which use software metrics like complexity and size. Another category is estimation models that predict reliability during development, while predictive models can be used early on. Common techniques involve stochastic models, reliability block diagrams (RBD), and fault tree analysis (FTA).

Model-based

- **Stochastic Models:** These models use statistical methods and historical failure data to predict future reliability. They can be categorized by their approach, such as reliability growth models that apply during testing or non-homogeneous Poisson process (NHPP) models, says IOSR Journals.

- **Statistical Models:** Examples include Autoregressive Integrated Moving Average (ARIMA) models, which are good for linear time-series data, and others that handle non-linear data.

- **Machine Learning (ML) Models:** These use advanced techniques like neural networks and support vector machines to predict reliability based on historical data.

Metric-based

- **Product Metrics:** These measure attributes of the software itself, such as complexity, size (e.g., lines of code), and functional points.

- **Process Metrics:** These assess the processes used during development, such as testing coverage and development effort.

Analysis techniques

- **Reliability Block Diagrams (RBD):** A graphical representation of a system that shows how components are connected and how their reliabilities combine to determine the overall system reliability.

- **Failure Mode, Effects, and Criticality Analysis (FMECA):** An analytical method to identify potential failure modes, their causes, and their effects on the system.

- **Fault Tree Analysis (FTA):** A top-down, deductive failure analysis that uses a tree-like structure to identify the root causes of system failures.

- **Failure Reporting, Analysis, and Corrective Action System (FRACAS):** A system for tracking failures, identifying trends, and implementing corrective actions.

- **Weibull Analysis:** A statistical method used to analyze product life data and predict reliability based on time-to-failure data.

4. IMPORTANCE OF SOFTWARE RELIABILITY PREDICTION

Everyone is aware of the necessity of software systems in all domains that affect every facet of society. This encourages the creation of software that doesn't fail. Software applications of this kind require labor-intensive techniques. Therefore, the creation of dependable software that accurately provides reliability is necessary. The economy and other elements are significantly impacted by software failures. The following factors, such as error prevention and defect detection, are examined to prevent software failure. There are already a variety of software metrics available to assess the software's dependability. Although there are several Software Reliability Growth Models available today, the projection has not yet hit its peak. Improved development, risk, and configuration management procedures, among others, can lead to increased reliability. These days, software failure is a common problem since it can't achieve the desired performance. The reliability element is anticipated at every stage of software development. Very little research is available for the post-delivery period, however many growth models are created for the evaluation or operating phases. It is challenging to determine whether software development processes have a greater impact on reliability because of bugs and flaws. Because every user experiences the software system in a different way or through a different module, the reliability measured through the user experience will range from person to person. To address these problems and examine and forecast a software system's reliability with some hypotheses, probability-based modeling has been proposed. Since every model is

different and based on software failure history, no single model can be applied to every situation. Each model has certain benefits and drawbacks. To assess the software system's dependability, a precise model must be selected based on the software system, development process, available input parameters, and presumptions. Nonetheless, the researcher has put out a number of models to evaluate the liability with few assumptions. In this field of study, creating generic models is still a difficult problem.

5. Critical Review of the Software Reliability

The review of literature is the procedure of summarizing, producing or assessing the material discovered through a research study. Subsequently, the number of application areas is extremely large, and the quantity of specific components is nearly endless. Since failure can't be prevented entirely, it is imperative to lessen the event's likelihood and the effect of disappointment when it happens. This is one of the essential parts of reliability analysis. It is assumed that imminent states are contingent only on the current state and not on past events. None of the researcher has been attempt in these area/parameters. Generally speaking, there are double categories of optimization problems: local and global optimization. The goal of local optimization is to locate the function's maximum or minimum value inside a limited area of its value interplanetary. In the current scenario in the context of global race and quick delivery system, it becomes fundamental aimed at all manufacturing plants to achieve optimally during their normal life span. This clarifies the developing interest for execution and study on the availability analysis in various manufacturing industries. An inclusive literature review linked to the situation of reliability productions is conversed.

Gorji (2023) studied on the Multi-objective approaches for problem optimizations in this field are further examined in the article. In view of upcoming trends in this quickly developing field, the considerable potential of meta-heuristic optimization techniques is emphasized as a crucial component in tackling these issues and enhancing overall sustainability and efficiency. Agrawal et al. (2021) introduced the first dualistic version of his GSK approach for chin assortment issues by incorporating binary subordinate and senior extraction and division stages. In standings of accurateness and the fewest attributes employed, this method outperformed the others.

To eliminate local optima in the sine/cosine algorithm, Abd Elaziz et al. (2017) suggested a hybrid strategy that makes use of the differential evolution algorithm's local search mechanism. Abetter-quality sine/cosine algorithm stoodestablished on eight UCI statistics sets, and it outperformed the others in rapports of power measurements and statistical analysis.

The latent and adaptability of evolution-based algorithms in resolving challenging optimization issues are demonstrated by these instances. When functional to UCI benchmark datasets via a KNN classifier, Emary et al. (2016) stayed the chief to device a binary type of the FFA, using a threshold assessment to provide efficient examination quality and quick solution identification. Poh et al. (2017) have provided a choice support system to develop an all-inclusive analytic net model that would explain the interdependency between the components in the current situation.

Tseng et al. (2016) this paper involved with sensitivity and dependableness analysis of a system that is governable having 'S' units of heat standby and 'M' units which are operative. Service times and failure times of standby units or in service units are alleged to go after the sharing which is exponential.

Ranjan et al. (2015) introduced a risk model base on gamma plus exponential failures. Foraging failures are linked to gamma failures, whereas accident failures are linked to exponential failures. Monte Carlo simulation and the Markov chain model were used for analysis. In contrast to an exponential distribution accidental failure curve, the model's curve will resemble a gamma model with aging failure predominating.

Park et al. (2021) an automobile or vessel's engine can be made lighter, which can increase engine performance and reduce CO2 emissions. As an end, this learning was done to determine how Si and Mg added to aluminum could advance the powered belongings of AC4CH alloy, a substitute for cast iron in locomotive construction. The Si edifice, grain enhancement, and heat behavior were refined to enhance the alloy's mechanical properties. A 300-hour durability test was also used to confirm the suitability of a cylinder block made through the changed AC4CH amalgam for a diesel locomotive.

Rathi et al. (2023) this study focuses on estimating steadfastness of a four-unit parallel cold standby organism. Phase-I unit repairs have been prioritized over phase-II unit repairs. However, there is no emphasis placed on how the units operate during any phase. There is only one repair facility that handles all kinds of system errors that may arise. Following repair, every unit functions like new and the switch devices are regarded as flawless. The units' failure time is taken to be constant, but their repair time is subject to an arbitrary probability distribution.

Thein and Park (2009) discussed the prerequisite for high reliability and availability because of current market demand. It was discussed that software failure is overwhelming compared to hardware failure because of numerous reasons, such as the complexity of software, large client size, and the nature of the application. Software rejuvenation utilizing virtualization ends up being extremely successful in dealing with software aging.

Shabbir, and Khalique (2024) the paper employs dependability estimation and parametric estimation techniques. To assess the effectiveness of the Generalized X Exponential Distribution (GXED) dependability estimator, a real data set is provided. Results demonstrate that the reliability parameter estimator fits real-world scenarios extremely well when the suggested distribution is used.

Mondal and Maiti (2023) in this work, we build an inspection plan for reliability acceptance sampling in order to make a decision about accepting or rejecting a large number of items whose lifetimes are governed by the OPPE family of distributions. There is infinite support for the OPPE distribution. There is discussion of the sampling process, operating characteristic curve, and plan design. It is mentioned how the plan parameters are determined using an algorithm. To safeguard the consumer's degree of confidence, the ideal sample size is established. The Lindley and the exponential are the two simplest specific options from the OPPE family that are selected as examples, and the optimal plan parameters are recorded and contrasted.

Ghamry et al. (2022) this study focuses on the mutually dismissed repairable equivalent k-out-of-n heartfelt standby organisation because failure is examined for its fuzzy consistency and accessibility. Numerical examples are utilised to clearly and analytically illustrate the fuzzy accessibility and consistency meaning of this uncertain organisation. This will be helpful in assessing some actual systems that may have defects that lead to malfunctions, like electrical systems, spacecraft navigation systems, and military systems. We can use the suggested approach to analyse more complex repairable systems in the future, like linear and spherical successive k-out-of-n systems by hot, emotionless, or standby with various fuzzy time-varying disappointment and reparation rates.

Zhang and Zeng (2016) have studied the maintenance modeling of different systems consisting of a number of non-identical units. They have told that when we apply modeling of maintenance on systems, then the probability of all maintenance activities play a key role in modeling.

Cheung (1980) talked about the markov reliability model was detailed by the assumption that steadfastness of module and between module control transfers is independent. Using sensitivity analysis, one can distinguish the modules that are critical to system reliability.

Vilkomir et al. (2008) examined the fault-tolerant computer framework with a few recuperation systems. Markov chain was utilized, yet it was confusing and required the utilization of software tools. The proposed model gives the diagnostic strategy to calculate availability. By this methodology, accessibility improvement could be applied to a framework.

6. CONCLUSION

The concept of software reliability entails measuring and improving reliability. A software product's approval or rejection is determined by its dependability. The authors propose generalized reliability prediction models that can be used at every stage of development. The time and expense required to apply various reliability prediction tools at various stages will also be decreased by this generalization. This study's primary goal is to inspire young researchers to build a new reliability survey framework that may be the most effective and appropriate for the current software development technique. Because every designed model has advantages and disadvantages, it may only be appropriate for certain projects and procedures. The goal was to provide a user an understanding of these models' utility so they could choose the best model for a particular software development environment. Software acceptability is confirmed during acceptance testing by creating inputs based on functional usage. Fault seeding is impractical during this stage, and disclosed flaws are typically not fixed right away. Thus, the theories of fault spreading and intervals between failures are irrelevant. User input might not be inconsistent during the working phase. This explains why a user could use the same software feature or path on a regular basis. In systems that operate in real time additionally, inputs may be agreed upon, removing their randomization. Furthermore, errors are not always fixed right away. Fault-count models are probably the most applicable in this setting and might be used to track software failure rates or figure out when a new release should be installed.

7. LIMITATIONS OF THE STUDY

Limitations of software reliability analysis prediction include the inherent complexity of software, the difficulty of accurately modeling and predicting human factors, and limitations in data quality and model applicability. Models may be based on flawed assumptions, be unsuitable for complex modern systems, or be prone to over fitting, especially when using machine learning techniques.

- **Inaccurate assumptions:** Many models make assumptions about the nature of software faults that do not hold true in real-world conditions.
- **Inapplicability to complex systems:** Traditional models often fail to adequately analyze the interactions between components in complex, modular systems.

- **Generalization issues:** Some models, especially older ones like Nonhomogeneous Poisson Process (NHPP), are only effective in specific cases and struggle to generalize to different software types.
- **Overfitting:** Predictive models, particularly those using machine learning, can overfit the training data, leading to poor performance on new, unseen data.
- **Dependence on data:** Many models, including deep learning, are highly dependent on the quality and quantity of failure data. Gathering this data can be challenging and time-consuming.
- **Human factors:** Factors like design quality, developer skill, training, and maintenance are crucial to reliability but are extremely difficult to quantify and forecast accurately.
- **Faulty input:** The accuracy of any prediction is limited by the accuracy of the initial failure rate data and other inputs, which can be subjective and incomplete.

REFERENCES

- Gorji, S. A. (2023). Challenges and opportunities in green hydrogen supply chain through meta-heuristic optimization.*Journal of Computational Design and Engineering,*1143–1157.
- Agrawal, P.,Abutarboush, H. F., Ganesh, T. and Mohamed, A.W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access,* 26766-26791.
- AbdElaziz, M. E.,Ewees, A. A., Oliva, D., Duan, P., and Xiong, S. (2017). A hybrid method of sine cosine algorithm and differential evolution for feature selection. *International Conference on Neural Information Processing,*145-155.
- Emary, E.,Zawbaa, H. M., and Hassanien, A.E. (2016). Binary grey wolf optimization approaches for feature selection. Neurocomputing, 371-381.
- Poh, K. L., and Liang, Y. (2017). Multiple-criteria decision support for a sustainable supply chain: applications to the fashion industry. *Informatics Multidisciplinary Digital Publishing,* 36-42.
- Tseng, C. Y., Wu, L. C., and Jia Y. C. (2016). Reliability and Sensitivity Analysis of the Controllable Repair System with Warm Standbys and Working Breakdown. *Computers & Industrial Engineering*, 84-92.

- Ranjan, R., Singh, S., and Upadhyay, S. K. (2015). A Bayes analysis of a competing risk model based on gamma and exponential failures. *Reliability Engineering & System Safety*, 35–44.
- Park, B. Y., Kim, Y., and Lee, K. (2021). Optimization Studies of AC4CH Material in the Cylinder Block of a Diesel Engine Application. *MDPI*, 9(1), 70-78.
- Thein, T., and Park, J. S. (2009). Availability analysis of application servers using software rejuvenation and virtualization. *Journal of Computer Science and Technology*, 24, 339-346.
- Shabbir, F. and Khalique (2024). Generalized X-Exponential Bathtub Shaped Failure Rate Distribution and Estimation of Reliability of Multicomponent Stress-Strength. *Reliability: Theory & Applications*, 1(77), 465-473.
- Mondal, A. and Maiti, S. S. (2023). Reliability Acceptance Sampling Plan For One Parameter Polynomial Exponential Distribution. *Reliability: Theory & Applications*, 3(74), 596-609.
- Ghamry, E. E., Muse, A. H., Aldallal, R., Mohamed S. (2022). Availability and Reliability Analysis of a k-Out-of-n Warm Standby System with Common-Cause Failure and Fuzzy Failure and Repair Rates. *Mathematical Problems in Engineering*, 1-10.
- Zhang, X., and Zeng (2016). Optimal maintenance modeling for system with multiple non identical units using extended DSSP method, *American Journal of Operations Research*, 6, 275-295.
- Cheung, R. C. (1980). A user-oriented software reliability model, Software Engineering. *IEEE Transactions on Software Engineering*, 118-125.
- Vilkomir, S. A., Parnas, D. L., Mendiratta, V. B., and Murphy, E. (2008). Computer systems availability evaluation using a segregated failures model. *Quality and Reliability Engineering International*, 24, 447-465.