RESEARCH ARTICLE

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
Vol. 04, Issue 10 October
www.rspsciencehub.com

Check for updates

http://dx.doi.org/10.47392/irjash.2022.061

Decision Model Based Reliability Prediction Framework

Nirsandh Ganesan[1], Nithya Sri Chandrasekar[2], Gokila[3], Varsha[3]
[1]Research & Design Engineer, KEDS GROUP R&D, Tamilnadu, Coimbatore, India
[2]Research Analyst, KEDS GROUP R&D, Tamilnadu, Coimbatore, India
[3]Research Intern , KEDS GROUP R&D, Tamilnadu, Coimbatore, India

Emails: nirshanthkodai@gmail.com, nithyachandrasekar2702@gmail.com, gokilaind@gmail.com, varsha18gopalan.@gmail.com

Article History

Received: 16 August 2022
Accepted: 26 September 2022

***Keywords**:*

Reiability;
Decision Tree;
Software;
Accuracyrate;
dataset:

Abstract

Under every situations, neither the specific pattern model could be used, despite of the extensive data analysis that attempts to expand reliability models of software. To make better use of current modeling techniques that are in usage such as combination and the model selection process, as a result, several other latest software reliability researches had been opted. Ineffective software reliability prediction is caused when incorrect model selection or weight allocation is trained frequently. This results in overrunning of the schedule. For combining various software reliability models on the keystone basis of multi-criteria decision trees, we postulate a methodical framework for prediction of reliability in this research paper. Based on experimental trends of multi-criteria sourced from multi reliability concepts, the method for model selection is suggested. For better and instantaneously allocation of weight per model, the decision tree with diminished defect edging ability deems the models with the predictive patterns preferred to be better. In this paper, investigation is done on the prospect of over- or under-prediction of the recognized models and the productive models in both predilected kind groups are weighted together. The proposed method exceeded current methods in terms of prediction accuracy rate, according to the results of analysis.

1. INTRODUCTION

A majority of Software Reliability Growth Models have been suggested to enhance prediction of the software reliability, over the last forty years. Software experts have worked hard to attain increased software reliability during the testing phase, as unreliable software can harm humans or a company's image. Maintaining oversight over the software testing process is about there the primary concern in this case, by predicting whenever a probability of failure will collapse underneath an accepted limit. The factor that plays a crucial part to the growth of this behavior is the accuracy rate of software reliability models. The realization of Software Reliability Growth Models (Dharmasena, Zeephongsekul, and Jayasinghe) has been confined, for a variety of purposes. The project data tends to vary greatly and frequently that seems unconfirmed to the fundamental assumptions of modeling techniques, also the experts have had no consistent method of deciding the framework in ahead of time. To evaluate the results of models on identified data in different factor, they frequently use multiple criteria. Experts are confronted with competing standards, and the comparative models precedence in each standard may transform as progression of testing, since modeling. In the software reliability application area, it is well understood besides that in a statistical hypothesis test performance to perceived data that in correlates with the prediction. Experts struggle to determine which model is far more prone to consider reliable predictions, in this situation. Previous works on prediction of software reliability has committed to building more comprehensive and possibly quite complex designs, instead of using current models quite proficiently in use. To obtain the optimal solution certain methods use multi-criteria data that limits their concentration on model prediction performance is the main consideration among experts. Multi predictive learning in brief overview is developing a framework enables for further enhancement of robust and accuracy rate of predicting software reliability by proficiently identifying and incorporating conceptual methods. This technique would be a significant concern for experts. To overcome the troubles in Software Reliability Growth Models, the automated methodology is suggested. The observations from various software reliability data sources utilizing prediction techniques for the analysis of software reliability, is also highlighted in this study.

2. PROVOCATIONS IN SOFTWARE RELIABILITY GROWTH MODELS

It has long been recognized that any model will not be effective for all situations, despite the increasing number of Software Reliability Growth Models (Dharmasena, Zeephongsekul, and Jayasinghe Amin, Grunske, and Colman). As a result, for a specific project multiple applicant Software Reliability Growth Models is considered. As Software Reliability Growth Models all seem to have underlying assumptions that varies and are frequently violated in training phase, so realizing which model features for advancement is complicated. The tough challenge that is generally directed by the software professionals experience is standard method to implement several Software Reliability Growth Models and to choose one based on evaluation metrics. For experts, than simply clarifying prior actions, accurate prediction of software's future conduct is preferred. As a result, an approach must be evolved that is reliable and that the experts could perhaps effectively implement and use for achieving prediction performance. Only after fitting the models to observed data, experts could assess the effectiveness of models in a range of methods.

2.1. Dependent Circumstances

The most frequently applied reliability condition is outlined in this chapter. Various conditions were used to evaluate the models, in this field of reliability. The elements of the model's performance that are different, is concentrated for each condition referred. We use the majority of the parameter used in each and every condition, with the exception of some closely correlated parameter, in this research. The descriptors below are identical. Mean Square Error is a metric to calculate the values that vary from the observed data. The lower the MSE, then relatively small the appropriate inaccuracy rate is obtained. Apart from Mean Square Error, the Mean Absolute Error attempts to measure the residual errors. This parameter is less attentive to the occurrence of estimation errors in Mean square error. The Rsquare value indicates the alignment that explicates the data variation. The estimated average errors among both the predicted values and revealed overall data are known as bias. The bias that is near to 0 implies that the assessment is free of bias. The more subsequent errors are reprimanded by the weighted least square error. The lower the weighted least square error, then the correlation to latest data is improved. The impression of risk of models is evaluated by Predictive Ratio Risk (PRR). The smoothness of the values obtained is measured by Noise. Mean Square Error, measures the curve's fitness level, whereas bias measures the curve's tendency for sliding on one side.

3. SOFTWARE RELIABILITY PREDICTION FRAME WORK BASED ON DECISION TREE

Based on multi-criteria model selection and combination, new method for predicting software reliability, has been introduced in this paper. To create more accurate multi-criteria software reliability [10-12] (Karunanithi, Whitley, and Malaiya Tohma et al.). predictions is the main objective of this method. The overall technique is depicted in Figure 1. To teach the empirical patterns of criteria, this method uses decision tree algorithms. A subset of the criteria (that are judged actually useful) is often incorporated in the trained decision tree model. Based on empirical evidences, the approach automatically identifies the model that is most likely to generate the most reliable predictions, instead of relying on experts' particularized judgments. By understanding the prediction behaviors from observational data sources, the Reduced Error Pruning Tree spot checks the design. Using an Alternating Decision Tree, the over- and under-prediction proclivities of the detected models are categorized. For a more accurate prediction of reliability, the models are integrated.

3.1. Designing the Decision Trees

The 2 tree-based techniques are being used in the proposed methodology to characterize the models using software testing data. The evidence-based requirements trends have been given training by Reduced Error Pruning Tree, which have an effect upon modeling predictive performance, and Alternative Decision Tree shows trends for over- and an under. It is a problem in bi classifications. Such tree techniques visualize the cumulative influence of circumstance (parameter) & help determine what factors were relatively more important over each function. It's indeed essential to obtain user's input related to a present and predict points (percent) against by the intended test for generating examples of such technique's moment. The models have been applied to observed evidence by professionals. Make predictions anywhere at point during scheduled test as for near future. Authorities next choose whether or not it should continue to test. According the client, the process involves breaking down every scientific data in training and validation groups input. It after every instance data has indeed been divided between training and validation subgroups, the modeling coefficients are determined just on testing sets as well as the system parameters just on testing sets a Non-linear Least - square evaluation of a training sample linear regression is a very well statistical technique. This method is widely employed for reliability modeling research. The comparative parameters (explanatory circumstance) are used after multiple models have indeed been given to a training sample (i.e., observed failure data). The training set is used to calculate the variables, whereas the testing subset is used to calculate the variables (Lyu and Nikora Khoshgoftaar and Woodcock). The variable is being used to construct the predictive parameter subset of test. The accuracy rate of prediction of the model is reflected by such parameters a number of designs. The proposed approach to learn Alternative Decision Tree uses Bias just on separate process as just a target attribute, and some comparative parameters related to excessive (or under-) fitting. As explanatory variables, utilize projection, bias is indeed, which measures whether prejudiced an individual are propensity is for prediction to tilt with one side. Those assessment methods focus on various elements of a model's effectiveness. A collection of measured data has been used to educate those occurrences before they could be educated. Inside a data, each one of the criterions is standardized. The min and max numbers to normalize numerical values inside a spectrum, normalizing is used, as well as the accompanying formula could be used.

3.2. Predictive Models Identification

They are using a regression strategy to find out how model predict the outcome. The numerical attributes were managed using Reduced Error Pruning Tree, as well as a regress is performed by trimming a graph. Data augmentation or variation is being used to build the architecture of the framework. Every data does have a theory which has been fit to that too. The structures of the competing assessment criteria are being used to assess whether likely a modeling would be to foresee the future failure mechanism of technology by using all instances from of the actual data. Reduced Error Pruning Tree rates every model on a large target following understanding the similarities.

The figure 2 explains the Alternate Decision Tree Model designed. Using all the events from the data sources, the classifiers are made for the classification technique.

4. EXPERIMENTAL FRAMEWORK

In this section, for testing the suggested technique the experimental research design is described.

The whole test method consists of three phases, as illustrated in **Fig. 3 .** To assess the findings using the leave-one-out cross validation approach, in Step 1 we use number of data sources. This strategy allows us to test the suggested approach's predictive accuracy on new datasets that we haven't seen before. While the remaining datasets serve as the training set, one dataset serves as the validation set. To evaluate the prediction performance in Step 2, the test

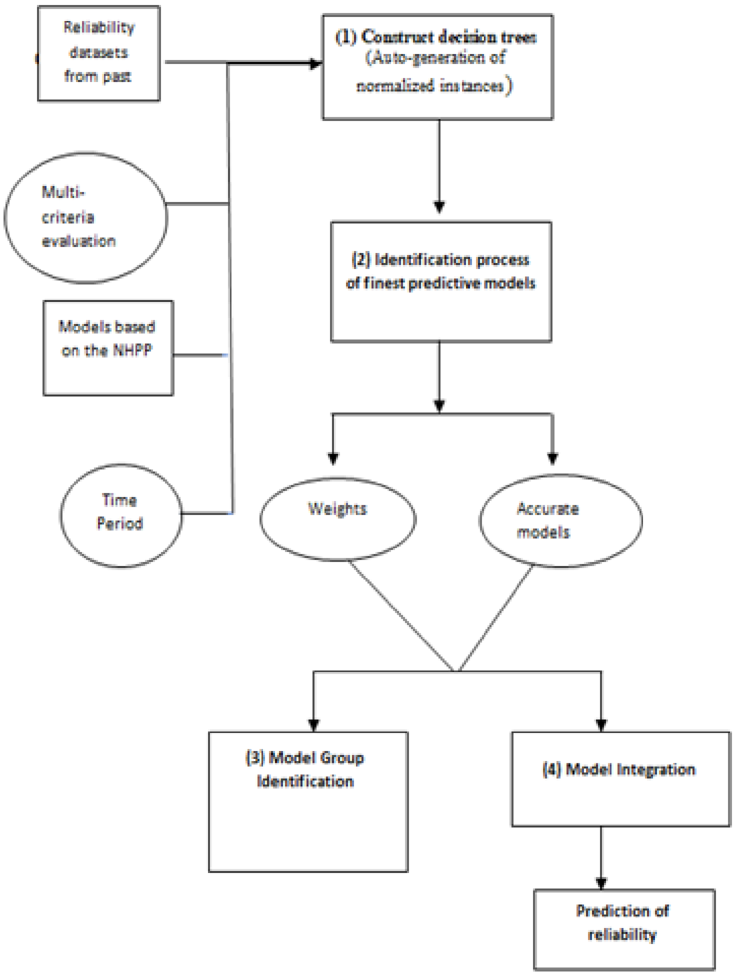


FIGURE 1. The overall technique

dataset is utilized, following the creation of classifiers from the training set. This process is done till all the samples have been evaluated at minimum once. The prediction results are evaluated, in step 3 using three prediction parameters.

The reliability data for about 40 applicants were utilized as research subjects. From a variety of sources, the datasets have been collected such as PDE, ML, LC, Apache Safe ZXing, Cyber Security & Information systems information Analysis Center and existing literature on software reliability, also includes information on failures in a software development project's testing procedure. The datasets are frequently been utilized as a reference point for software reliability models. The data was primarily obtained through rigorous controls, during a system test. The datasets are mentioned in the table below.

TABLE 1. Number of datasets for the purposes

PURPOSE	DATASET
OSS	1 - 9
OS	10 - 14
Command control	15 - 22
Real time control	23 – 26
Word & Signal processing	27, 28
Network	29 - 32
Clinical analysis	33 - 37
Administering software	38 - 40

5. RESULT & DISCUSSION

The experimental findings and analysis of our decision tree-based architecture and other method's prediction performance, is outlined in this topic. In terms of prediction performance, our strategy is compared to various typical model selection, augmentation and data-driven strategies. The first step is to compare three distinct model evaluation meth-

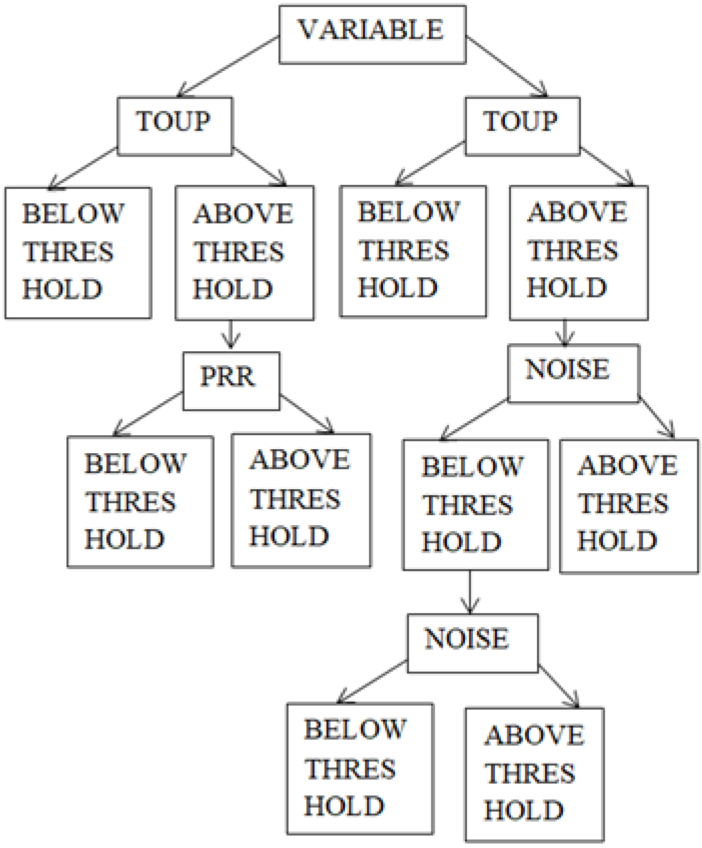


FIGURE 2. Designing of AlternateDecision Tree Model

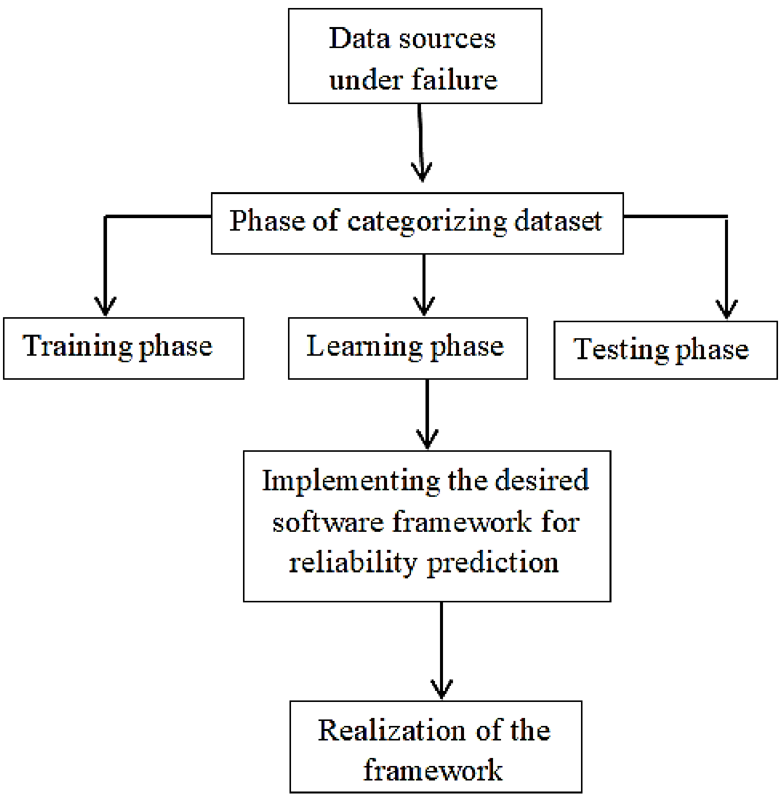
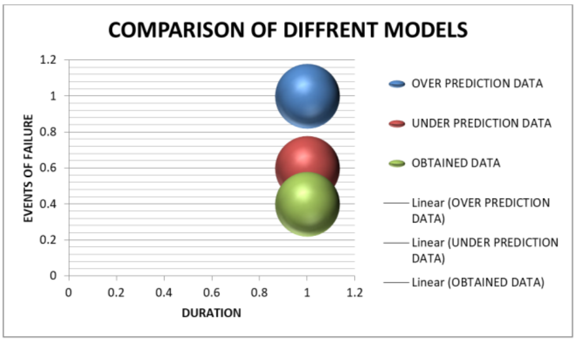


FIGURE 3. The desired methodology

ods; the Nave approach, the sum of circumstance ranks, and the distance based methods are compared; the Dynamic Weighted Combinational Model is also compared to our method, which is one of the typical augmentation models.



Graph 1: Comparison of different prediction techniques

The above graph best, describes that the proposed methodology has achieved the better performance in comparison to the other techniques.

Based on Support Vector Machine for Regression, (Elomaa and Kaariainen Tian and Noore Moura et al.) we use the Data-Driven Software Reliability Models, which is a series data analysis, since several recent studies show that this model outperforms other techniques in software reliability prediction. To anticipate Data-Driven Software Reliability Models (Zhou and Leung Makridakis et al. Zhao and Xie), we utilize a recursive technique for a long term purpose. To evaluate the predictive accuracy of techniques, we utilized the cross validation technique on all datasets utilized in this work. We saw that as testing proceeded, the Mean Error of Prediction values of approaches decreased. It makes sense since the larger the prediction horizon, the greater the uncertainty and the more data needed for training. The Mean Error of Prediction tests on the forty software error datasets show that the method is more reliable and resilient than the others in the vast majority of testing periods (i.e., in long-term or short-term).

5.1. Limitations of Proposed Method

In this report, the constraints of our study are described. When we try to apply our study data to real-world situations, it seems an initial challenge to reality. The results may not be typical, as this report looks at forty cases. It implies we can unintentionally choose solutions with certain models that provide higher outputs. We gathered and evaluated data source that had previously been frequently applied in research, to avoid the above issue. In this research, extending the methods and information applied is also possible. More reliable outcomes can be achieved from Tree based methods that have been developed on a larger number of documents. As a result of the properties of our technique, which employs scientific findings, we may anticipate improved predicting with some more records. To create more trustworthy decision trees and ensure correct predictions more records should be collected. In this paper, study's circumstancing parameter and assumptions will act further as elements. Further relevant parameter and concepts may exist, to enhance quality performance of prediction; all parametric condition can be included into our system reliability projection architecture and contribute significantly.

6. CONCLUSION

In this study, we present a reliability methodology of prediction by employing, decision tree-based system. We developed sub models using obtained results, to describe the current Software Reliability Growth Models. In the developed decision tree system, only a sample of the factors determined really useful for prediction will be incorporated. To produce the most valid inferences based on actual data instead of operators subjective judgments; this method immediately gives a value to each concept and presumes the model that is most likely. To enhance dependability projection even more, the system includes the discovered designs with account of the given values and the percept of above or below. Using previous records automation of the computing of parameter and the creation of decision trees, thus we have created a program that allows the foundation. In trials utilizing various test records of number 40, the suggested methodology performed well, indicating that it is considerably highly effective and resilient in predicting system dependability in the far and near range. We can hope improved results of prediction, as the skilled decision tree algorithm may be utilized in various evaluation of proposed projects, and with additional samples.

References

Amin, Ayman, Lars Grunske, and Alan Colman. "An approach to software reliability prediction based on time series modeling". *Journal of Systems and Software* 86.7 (2013): 1923–1932. 10.1016/j.jss.2013.03.045.

Dharmasena, L Sandamali, P Zeephongsekul, and Chathuri L Jayasinghe. "Software Reliability Growth Models Based on Local Polynomial Modeling with Kernel Smoothing". *2011 IEEE 22nd International Symposium on Software Reliability Engineering* (2011): 220–229. 10.1109/ISSRE.2011.10.

Elomaa, T and M Kaariainen. "An Analysis of Reduced Error Pruning". *Journal of Artificial Intelligence Research* 15 (2001): 163–187. 10.1613/jair.816.

Karunanithi, N, D Whitley, and Y K Malaiya. "Prediction of software reliability using connectionist models". *IEEE Transactions on Software Engineering* 18.7 (1992): 563–574. 10.1109/32.148475.

Khoshgoftaar, T M and T G Woodcock. "Software reliability model selection: a cast study". *Proceedings. 1991 International Symposium on Software Reliability Engineering* (1991): 183–184. 10.1109/ISSRE.1991.145378.

Lyu, M R and A Nikora. "CASRE: a computer-aided software reliability estimation tool". *[1992] Proceedings of the Fifth International Workshop on Computer-Aided Software Engineering* (1992): 264–275.

Makridakis, S, et al. "The accuracy of extrapolation (time series) methods: Results of a forecasting competition". *Journal of Forecasting* 1.2 (1982): 111–153. 10.1002/for.3980010202.

Moura, M Das Chagas, et al. "Failure and reliability prediction by support vector machines regression of time series data". *Reliability Engineering & System Safety* 96.11 (2011). 10.1016/j.ress.2011.06.006.

Tian, Liang and Afzel Noore. "Dynamic software reliability prediction: an approach based on support vector machines". *International Journal of Reliability, Quality and Safety Engineering* 12.04 (2005): 309–321. 10.1142/S0218539305001847.

Tohma, Y, et al. "Hyper-geometric distribution model to estimate the number of residual software faults". *[1989] Proceedings of the Thirteenth Annual International Computer Software & Applications Conference* (1989): 610–617. 10.1109/CMPSAC.1989.65155.

Zhao, M and M Xie. "On the log-power NHPP software reliability model". *[1992] Proceedings Third International Symposium on Software Reliability Engineering* (1992): 14–22. 10.1109/ISSRE.1992.285862.

Zhou, Yuming and Hareton Leung. "Predicting object-oriented software maintainability using multivariate adaptive regression splines". *Journal of Systems and Software* 80.8 (2007): 1349–1361. 10.1016/j.jss.2006.10.049.

Embargo period: The article has no embargo period.

To cite this Article: Ganesan, Nirsandh, Nithya Sri Chandrasekar, Gokila, and Varsha. "**Decision Model Based Reliability Prediction Framework**." International Research Journal on Advanced Science Hub 04.10 October (2022): 236–242. http://dx.doi.org/10.47392/irjash.2022.061