A Survey on Software Reliability Models used for Quality Improvement of Software using Machine Learning Techniques.

Jullius Kumar[1], Dr. D. L. Gupta[2]
M Tech (Student) [1], Associate professor[2]
Department of Computer Science and Engineering, KNIT, Sultanpur, INDIA

Abstract: Quality of software demands for highly reliable software to evaluate the failure rates along the time with defined environment. The probabilistic approach which measures the occurrence of failure free software randomly. In the real time environment how model used to resolve the problems to predict the reliability of the available data (failures or faults). No single model is suitable in all circumstances. In this paper, several proposed model or a methodology that has been studied which is used for predicting the reliability of software and forecasting. A comparative analysis has been performed to know the each reliability model. We have also discussed the various machines learning algorithm as Decision Tree (DT), Fuzzy Inference system (FIS), Cascade Neural Network (CNN), Support Vector Machine (SVM), Artificial Neural Network (ANN), Bayesian Belief Network (BBN), and Naïve Bayes (NB). The various models have proposed their own technique and approach to resolve the failure free software or system. The other sections of the paper include introduction, related work, learning algorithms and conclusion with references.

Keywords: Forecast, Machine Learning, Prediction, Reliability, Software failure.

Introduction

In the life cycle of development of software the main focus is how to measure performance of system. Reliability of software is dynamic and stochastic, differs hardware reliability. This approach is failure free software operations of program for specified period of time in specified environment. [14]

The machine need quality that depends on monitoring of critical control leads safety and high software reliability in great demand. The managers and practitioners are to decide the paradigm is more beneficial and to what degree, in specific domain. Basically the architecture of client-server is presentation, business and database which are stored at back-end. The analytical model is described at specific testing process to predict software reliability in realistic environment. Different failure datasets can be collected for the predicting the software failure for standard projects. [12]

When checking whether to correct errors and subject to restriction, how much time will be spent on tests, and how much more difficult it is to fix an error during the operating process than the evaluation period. We should basically focus when to stop testing process when to delivery of software to end user by developing software cost model with risk factor which ensures the delivery of software product attaining predefined efficiency standard and cost minimisation and statistical NHPP model. [13]

In real time environment how model used to resolve the problems of failure system. The paradigm suggested such as the S-shaped, hyper-exponential form, the S-shaped paradigm of inflection. In real time projects the S-shaped pattern is more commonly observed. It is caused by failures or faults. During a test, have test cases run, observe results of runs, defect failures. After failure detection, failure analysis or finding fault, and fix fault must define. [15]

To solve time series and non-linear regression problems, the method SVM based on statistical approach. The parameters of SVM are determined by genetic algorithm. Although evaluating theoretical models to other models to forecast is more accurate to estimate and dependent on the data scale for failure. The two basic types of data are time domain data and the interval domain data (both input). Time domain data describe recording the separate time when failure occurs and the interval domain data is described by counting the cumulative failures occurred specific duration of time. [8]

Different models have various capabilities for calculating the precision to predict the specific test phases; no single model is focused on similar models. Various networks models and training regimes used for predicting the software reliability. The scaled representation of Input/output variable to give the better results (or accuracy) than binary encoded representation. The least square error minimization technique in estimating the parameters of models for significant data points.[9]

Some well known machine learning(ML) algorithm as Decision trees(DT's), Fuzzy inference system(FIS), Cascade neural network(CNN), Support vector Machine(SVM), Artificial neural Network(ANN) has been proposed to evaluate the performance of system to predict the reliability of software which has been measured by mean absolute error(MAE), root mean squared error(RMSE) and the precision. Precision is very effective with varied real life failure data sets. Failure Cumulative number of failures detected during testing period as input data, day of failure as output data. Arranging the data set into k-folds used for model training and validation. Various parameters are set for achieving the goal, validating the model using k-fold data to refine it and implementing it. [16]

Literature Survey

Hu et. al. Proposed models of reliability of software used to measure data from software errors. The robust model of neural for connectionist method is less restrictive. The Elman's persistent neural network is explored and forecasting software failures. Non-homogeneous Poisson Process(NHPP) and models Connectionist are the methodologies used. Specific predictive models have emerged in multiple situations with no common standard. The output of predictive errors from multiple neural network models is higher than, at least comparable to, NHPP development models. In the modeling framework of dynamic reliability progression, a neural network is used for progression prediction models and the creation of its own model of the failure cycle. The concept with several parameters gets more complicated as the basic size grows.[7]

Ron Kohavi proposed a model that suggests the decision tables which are the easiest spaces of inference, quick to grasp and supervised learning algorithm. Datasets consist of Continuous features some involve features and others have value-containing features. Data-set specified to validate while using incremental cross-validation, leave one and ten cross independent of the amount of folds selected taken at a time. To approximate the accuracy increasing node comprising tenfold cross validation. Cross-validation is replicated before standard deviation on mean went below one percent due to large uncertainty of the calculation. We did an algorithm for defined training and test set for data sets. We do ten-fold cross validation for the remainder of the data set around DTM inducer (Decision Table Majority), IDTM. The most vulnerable aspect of the IDTM algorithm is the calculation of the precision. As the search is always a heuristic quest, it will not be feasible to locate the right subset of features and that is not the absolute upper limit. Thus the upper bound does not attain precision provided; success above this point is unlikely without modifying the space of the hypothesis. For humans, the decision table is simple to grasp, especially if not too high. Some datasets we encountered are constant values; IDTM's estimation accuracy is contrasted with C45. Decision table doesn't imply the induction algorithm has strong hypothesis. It is difficult to generalize without a prejudice, to approximate the results of cross validation accuracy. Selecting the features

subset for IDTM to biased. The concept of gradual cross validation and deletion operations was also conceived, for number of cases, number of functions, number of label values.[10]
Rana et. al. proposed a system for major software enterprises, utilizing ML methods testing the ISO / IEC 15939 knowledge standard. We no longer need to know the exact relationships between baseline and derived measurements that create detailed model of study of how specific quality sub-characteristics influence higher quality or overall output characteristics In this context, the bottom-up methodology for quantitative evaluation of high-order quality characteristics for ML techniques such as Artificial Neural Network (ANN) to predict the quality category during development, in particular software modules. In quantitative assessment or qualitative class labels (such as high or low) values for different characteristics of high order which affect overall characteristics. This model is based on a process of measuring knowledge with top down approach.[1]
Kumar et. al. proposed a conceptual paradigm for the connection to operating device functionality. For improve functionality of the applications, data oriented methodology and rational approaches are used. A Bayesian Belief Network(BBN) methodology that underlies a device or dataset relationship. This reflects a graphical structure that offers a directed acyclic graph (DAG) context for inference and the prediction. For this model BBN has five indicators known as Techno Complexity, Practitioner Grade, Development Efforts, Audit Effort, and Urgency. Template is experimented with three project requirements and real failures obtained by applying BBN estimating the amount of faults predicted.[3]
Hoi et. al. Proposed a model based on a paradigm that establishes a modern quantitative metric to evaluate the quality and output of the software process systematically and to evaluate the process of defect management carried out in four real industrial software projects. For this article, the solution to machine learning is to use controlled classification techniques to define each task execution ”normal” or “abnormal”. Abstract raw data, held in database databases, preprocess data by converting specific data to construct the sequences, creating a particular sequence classifier, marking data sets, and testing the classifier. Usage of the sequence classifier i.e., “natural” or “abnormal” training classifiers uses ML algorithms to determine the output at the last stage. We’re following three well established and descriptive strategies in Machine learning such as naïve bayes (NB) algorithm, Support Vector Machine (SVM), Decision Trees (DTs).[4] Seliya et. al. suggested the CART-LS(Least Square), S-PLUS, and CART-LAD(Least Total Deviation) machine fault prediction model for regression algorithms being used for systematic assessment. Tree models that are used in design metrics to predict the amount of module faults. The methods used, such as Creating tree regression models, performance metrics, and average absolute error (AAE), Average relative error (ARE), etc. A comprehensive evaluation of currently available regression tree algorithms for software quality modeling is used in this article.[2]
Pattanayak et. al. proposed an early identification work on anticipated fault prone software modules in the software development process is critical to saving effort involved in the software development process is critical to saving effort involved in the process. Various ML techniques such as Fuzzy logic, Neural network, Bayesian model etc. used for prediction of software quality. ML methods include Neural Network, Bayesian Network, Genetic Algorithms, Fuzzy Logic, Tree-based Algorithms, Decision Tree Algorithms, and Revised CRT Algorithms. Nearly all strategies protected following specific observation points. Firstly, software quality relies on internal quality attributes which can be identified during software creation and external quality attributes which can be identified during software implementation. Secondly, the software quality assurance concept is focused on

the assumption that the external quality attributes are also a feature of internal quality attributes.[5]

Costa et. al. proposed a model to many conventional and non-parametric models, the method adopted is genetic engineering and boosting, which is used to increase performance. This method offers performance change over traditional genetic engineering which needs ten times the amount of executions. Reliability is an essential feature of product efficiency. Probability the program operates in a certain environment without faults for a period of time. Modeling device efficiency is less costly methodology. Classical genetic programming models are focused on period during which boosting strategy uses ten-fold booting iterations and software combination of models. Other model focused on scope, twenty GP models are trained in one feature range using twenty separate seeds for each structural criterion. Genetic programming boosting (GPB) approach and $(\mu+\lambda)$ GP method provides greater performance than traditional genetic programming utilizing just feature sets and is comparable to lower and identical expense solution to GP. The GP solution $(\mu+\lambda)$ provides stronger outcomes than traditional GP and GPB. This provides stronger outcomes for Standard error, median variance, receiving normal biased values.[6]

Jung-Hua Lo Proposed an early phase testing model for stable applications obtained which offers a stronger approach to problem solving (failure data). Different forms of input data used to forecast output models are chosen. Approach SVM models focused on Genetic Algorithms. SVM is a process focused on a mathematical approach to solving problems using the nonlinear analysis and time series. Genetic algorithms are used to evaluate SVM parameters when considering certain prediction models. The proposed model is more reliable for estimation, and less contingent on the scale of the outcomes of the fault. SVM is used to map training data into a higher-dimensional (non-linear function) and one-dimensional (non-linear prediction) space for the processing and resolution of linear knowledge in the feature domain. Software reliability growth models(SRGM) provide two simple input data types, first defining time domain data by reporting the specific period when errors arise. Second, the data about the interval domain is represented by counting overall number of faults that occurred during a specified time span. It took 28 encountered consecutive errors that were reported to check the software, 234 errors were found during consecutive execution-time errors.[8]

Karunanithi et. al. proposed a separate model with various predictive capacities in multiple test phases and no particular model depended on specific forecasts. The Connectionist paradigm examined different types of network regimes and form of data representation. Five well-known development models of program reliability use real data sets and are linked to each other for improved predictive accuracy. It uses neural networks and obtained results for predicting responsiveness of applications. The estimation of device stability may also be extended to feed forward networks (FFN) and recurrent networks to different training regimes. Neural network predictive precision utilizing approaches for representing data often analyzed. Various training regimes such as generalization training, prediction training, and teacher forced learning generalization training, teacher forced learning prediction training. Various model networks and training regimes are used to accurately simulate the results. The normalized Input / Output variable representation provides improved outcome (or accuracy) than the encoded binary representation. Overall, the Jordan net generalization model is effective at the end-point prediction and Jordan net regression models are strong at the next phase regression.[9]

Lee et. al. proposed a three multi-step forward prediction approaches for data-driven program reliability models, and compares their predictive success with data counting failures and time duration failures. Recursive approach allows greater estimation of fault tracking data relative to the time between fault results. The outcome demonstrates the applicability, also in long-term estimation, of data based processes. Identifying a good multistep forecast approach is beneficial. Software Stability Growth Models (SRGMs) are classified into two types of model Failure Count(FC) and period between model Failure(TBF) based on the form of failure data. The FC model focuses on number of errors in time period and the TBF models use as a random variable duration between failures. The three approaches use the correlation between past and potential data to forecast a multistep forward. (a)Recursive strategy (b)Active strategy (c)DirRec(Combines Recursive and Active Long Term Prediction Strategy). The different data initiatives, FC and TBF datasets were tested on all three criteria. The total difference between the expected value and real end-point average results defines how effectively the methods are forecasting the real results. The percentage of actual relative error and estimation of endpoints implies that in long-term projection, model precision is higher. Long-term estimation implies more volatility, forecasting multistep for device reliability is a rather challenging task.[11]

Kumar et. al. proposed various stages of computational model research used to assess the functionality of applications in a specific setting. This is where we concentrate on input back propagation algorithm and network design problems. ANN consists of brain containing many nodes that are similar to nerves. Node has a feature that is associated with it and sets local parameters to decide the node performance for given set of input. The network of neurons may be either single or multilayer-networks. The two large SRGM groups are parametric and non-parametric types. Most parametric models are technically stable based on NHPP. We test the measure of the proposed model utilizing parameters such as variable-term predictability, root-mean squared error(RMSE), mean absolute error(MAE), means error and standard deviation. Feed forward neural network(FFNN) using back-propagation algorithm used to effectively and significantly predict software reliability development, comparison of NHPP-based SRGM and other. In a practical context we use models to use connectionist methodology. Nonlinear trends and statistical failure results produce less predictive errors theoretically by understanding more about back propagation learning algorithm.[12]

Machine Learning

The In machine learning, data plays a significant role and the learning algorithm is used to discover and gain from the data knowledge or properties. Dataset consistency or quantity may influence the efficiency of the learning and prediction. "Machine learning," which highlights the significance of evidence in machine learning? The class dataset for dataset notation before moving straight into machine learning, we first define the dataset notation in a segment. There are two common forms of datasets. One is labeled and second is unlabeled:

Labeled dataset D describe $X=\{x^{(n)} \in R^{d}\}^{N}_{n=1,}$ $Y=\{y^{(n)} \in R\}^{N}_{n=1}$
Unlabeled dataset D describe $X=\{x^{(n)} \in R^{d}\}^{N}_{n=1}$

There are typically two forms of algorithms on the continuous problem and collection of datasets as supervised learning and unsupervised learning.

Supervised learning

The supervised learning given as the labeled dataset for training. Supervised learning aims to establish the relationship between selection of attributes and selection of labels, which is the knowledge and property we can obtain from the dataset called. The learning problem is denoted by grouping correspondingly with each function vertex x to be named. On the other hand, if those vectors x function correlates to a real value like R; learning problem is defined as regression. Often input from supervised learning research is used for prediction and recognition.

Unsupervised learning

Collection of training for unrestricted learning is Unlabeled dataset. Unsupervised learning is aimed at clustering, estimating the probability values, defining association among other functions, and decreasing dimensionality. Overall, an unsupervised algorithm would recognize more than one of the aforementioned properties at the same time, and the results of unsupervised learning on controlled learning can be collected.

The various supervised and unsupervised learning algorithms have been discussed as:

Decision Tree (DT)

The algorithm DT used for the regression as well as classification strategies used for problem solving. The decision tree's aim is to build a training model that can be used to predict the class or meaning of the target variable through learning from previous data (training data) by basic decision rules. In DT’s we continue from the tree's root to determine a record class label. We equate attribute roots of the value with those of a database attribute. Based on contrast, we obey the branch that corresponds to the value and jump to the next node.

Fuzzy Inference system (FIS)

A FIS includes a variety of laws, a range of input and output variables. Where Fuzzy Inference Method is used, the input and output variables may be multiple fuzzy collections. During operation, input variables use the computer input values to equate their sets, that is, they determine the degree of belonging of that input value to all the fuzzy sets of the network. The law related to the output variables relates to a degree; the value of the feature is determined by the whole effects. Fuzzy rules are fixed in motion. The input-output vector scheme for fuzzy inference and Fuzzy defined laws. Both variables that provide the input-output that contains the inference method array of fuzzy sets. The values representing variables are near; the rules are vague and are dealt with accordingly.

Cascade Neural Network (CNN)

It is similar to the class of neural network feed-forwards. But it is linked to the following layers with feedback from the previous layer. In a network it has three layers, next to secret layer, the output layer is explicitly linked with the input layer. As with feed-forward networks, provided enough secret neurons, a cascade-network of two or more layers will arbitrarily recognize some finite relationship Input-Output. Cascade-forward neural network to any sort of data may be used to describe the outputs. The downside of this method is that it accommodates the nonlinear correlation between input and output, while not eliminating the linear association between the two. In this analysis we extend the network in time series field [17].

Support Vector Machine (SVM)

The algorithm supervised learning that can be used in both regression and classification problems. However, it is still used in protocol problems. The algorithm SVM, we map each data structure must have value for specific coordinate as a n-dimensional space as point with n represents the number of features containing. The SVM classifier is a frontier preferably separating the two classes (hyper-plane / line).

Artificial Neural Network (ANN)

An ANN initially goes through a preparation phase where it learns to recognize the specifics of patterns, whether physically, aurally or textually. The network contrasts the real performance achieved during this controlled process which is intended to produce — the ideal performance. Using back propagation the disparity between all results is balanced. That ensures the network operates backwards, moving from the output device to the input units to change the weight of the links between the devices before the gap between the real and expected effect generates the lowest error possible.

Bayesian Belief Network (BBN)

The structure reflects the graphical method of directed acyclic graph (DAG) offers an inference and prediction context. Nodes reflect the random variables with a certain meaning possibly. The five criteria to be known as technical sophistication of this model, which requires the usage of development in applications. Level of Practitioner who explains the expertise of the substance with practice. Creating efforts requires the obligation that paper creation, process design and coding step depending on individual hours. Effort analysis, urgency defines the percentage of time compression.[3]

Naïve Bayes (NB)

Naïve Bayes Theorem's classification method, with an expectation of predictor independence. Simply stated, a Nãive Bayes classifier assumes that there is no relation between a particular feature being included in a class and any other feature present. We will calculate the likelihood of A occurring using Bayes theorem, if B has happened. Below, Argument B and Theory A. The premise here is to separate the predictors / functions. That's one particular item getting used that doesn't influence the other. And that is called naïve.

Conclusion and Future Scope

Software reliability is one of computing key qualitative dimensions, any software error made unreliable. In this paper we have reviewed various growth models of reliability to predict the reliable software for failure data. The paper discussed about the techniques of ML algorithms which discussed the supervised, unsupervised and reinforcement algorithms to forecast the software quality. In the article a comparative analysis of different models was discussed, and the techniques of machine learning (ML) algorithms were used to test system efficiency. The more study within the field is to extend and reduce the effectiveness of predictions and refine the findings for potential studies.

.

References

[1]. Rakesh Rana ; Miroslaw Staron "Machine Learning Approach for Quality Assessment and Prediction in Large Software Organizations", **DOI:** 10.1109/ICSESS.2015.7339243.

[2]. T.M. Khoshgoftaar, N. Seliya "Tree-based software quality estimation models for fault prediction", **DOI:** 10.1109/METRIC.2002.1011339.

[3]. Chandan Kumar ; Dilip Kumar Yadav, "Software quality modeling using metrics of early artifacts", **DOI:** 10.1049/cp.2013.2285.

[4]. Ning Chen, Steven C. H. Hoi, "Software process Evaluation: A Machine Learning Approach", Xiaokui XiaoPublished 2011, Computer Science, 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011).

[5]. Saumendra Pattnaik; Binod Kumar Pattanayak, "A Survey on Machine Learning Techniques used for Software Quality Prediction", DOI: 10.1504/IJRIS.2016.080058

[6]. Eduardo Oliveira Costa, Aurora Trinidad Ramirez Pozo, Silvia Regina Vergilio," A Genetic Programming Approach for Software Reliability Modeling", **DOI:** 10.1109/TR.2010.2040759.

[7] Q.p. Hu ; Y. S. Dai ; M. Xie ; S.h. Ng, "Early Software reliability prediction with extended ANN model ", **DOI:** 10.1109/COMPSAC.2006.130

[8]. Jung-Hua Lo, "Predicting Software Reliability with Support Vector Machines", **Published in:** 2010 Second International Conference on Computer Research and Development, **DOI:** 10.1109/ICCRD.2010.144.

[9]. N. Karunanithi ; D. Whitley ; Y.K. Malaiya, "Prediction of software reliability using connectionist models", **Published in:** IEEE Transactions on Software Engineering (Volume: 18 , Issue: 7 , Jul 1992), **DOI:** 10.1109/32.148475

[10]. Ron Kohavi, "The power of Decision Tables", **Publication:**ECML'95: Proceedings of the 8th European Conference on Machine Learning April 1995 Pages 174–189, https://doi.org/10.1007/3-540-59286-5_57.

[11]. Jinhee Park ; Nakwon Lee ; Jongmoon Baik, "On the long-term predictive capability of data driven software reliability model: An empirical evaluation", DOI: 10.1109/ISSRE.2014.28

[12]. Yogesh Singh ; Pradeep Kumar, "Prediction of Software Reliability using Feed Forward Neural Networks ", **DOI:** 10.1109/CISE.2010.5677251.

[13]. Pradeep kumar, Yogesh Singh, "Determination of Software Release Instant of Three-Tier Client Server Software System", Published in International Journal of Software Engineering (IJSE), Volume 1, Issue-3.

[14]. Aasia Quyoum, Mejraj-ud-din dar, S.M.K. Quadri,"Improving Software Reliability using Software Engineering Approach- A Review", International Journal of Computer Applications 10(5),DOI: 10.5120/1474-1990.

[15]. M ohba, "Software reliability Analysis Model", **Published in:** IBM Journal of Research and Development (Volume: 28 , Issue: 4 , July 1984), **DOI:** 10.1147/rd.284.0428

[16]. Pradeep kumar, Yogesh Singh, "An empirical study of software reliability prediction using machine learning techniques", September 2012, International Journal of Systems Assurance Engineering and Management 3(3), DOI: 10.1007/s13198-012-0123-8

[17]. Budi Warsito, Rukun Santoso, Suparti and Hasbi Yasin,"Cascade Forward Neural Network for Time Series Prediction", Published under licence by IOP Publishing Ltd.