Implementation of Machine Learning Techniques in Software Reliability: A framework

Manu Banga
Ph.D. Scholar ASET
Amity University
Noida, India
manubanga@gmail.com

Abhay Bansal
HOI ASET
Amity University
Noida, India
abansal1@amity.edu

Archana Singh
Professor ASET
Amity University
Noida, India
archana.elina@gmail.com

***Abstract*—In this paper review of existing literature in the field of software reliability models based on machine learning techniques presented. Software reliability is very useful tool in determining the software quality. By using machine learning techniques for getting unhidden parameters affecting software fault prediction for exploring various parameters leading to obsoleteness of software by presenting category of papers of software reliability, software fault prediction, software trustworthiness, software reusability, using machine learning techniques based on statistical inferences which could predict useful pattern on hidden data of faulty software database of empirical datasets related to software testing. After studying plenary relevant papers on faults generated during fault removal, faults already present, we proposed a novel approach based on identifying most relevant parameter affecting the software reliability using Machine Learning Techniques.**

***Keywords*—Software Reliability, Intelligent Software, Machine Learning Techniques,Faults, Failures, Feature Selection**

I. INTRODUCTION

Software is facing strong threats in the reliability, maintainability urging companies to develop intelligent software for enhancing trustworthiness of software thereby controlling failures. For that authors, implemented various Machine Learning algorithms for getting solutions in controlling parameters affecting the most [2][3]. In the modern world testing techniques and software reengineering critical and most crucial in determine software usability [6][8] Software usability is defined as the optimal use of software without failure under specified condition and time [9]. For developing intelligent software various techniques are available Information Retrieval, Web Mining, Artificial Neural Networks, Fuzzy Set Theory, Rough Set Theory, Artificial Intelligence [15]. Some Faults are originally present in the dataset, some faults generated during fault removing, may lead to failure of a complete system. As intelligent software becoming necessary formed by converging machine learning techniques on company faulty dataset for building reliable model in various dimensions like defense, banking, railways based on Early Prediction Model [7],is syndicate for software reliability prediction by maintaining low threats to designed software, according to A^3 framework Figure 1 of software usability Application, Algorithm and Architecture for reliable working of software without failure in real time environment. Among all the software reliability models Non-Homogenous Poisson process based on S-shaped, concave curve extensively used during our study for classifying software-debugging phenomenon. As software failure phenomenon causes monetary losses, time loss, as faults transform to deadly failures [12] leading to loss of information. Thus controlling faults at appropriately at the release time and careful examine them in a testing/debugging phase by using previous data of software failure for estimating defects remained under testing phase. Based on Failure History, Optimal handling of the defects mean function$m(t)$,and software intensity function $\lambda(t)$ of software reliability models validation is accessed to estimating &predicting defects remained in it [4][5]. Machine Learning plays a very crucial role in the knowledge discovery in the reliability assessment of software. It is applied for searching the hidden accuracy in the software usability in real time scenario. Various Machine techniques are applied for forecasting the software validation.

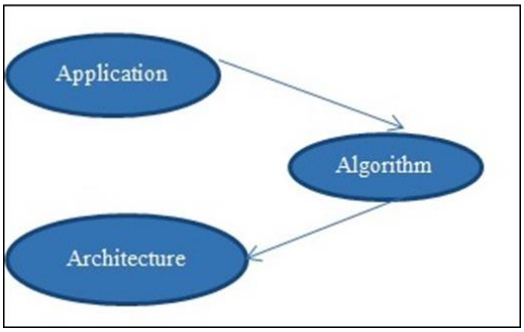

Fig. 1. A^3 Framework of Software Usability

Software Application are business application at user level at topmost framework providing GUI to user, Algorithm is sequence of steps on which input of user processes from database retrieval located at Architecture level where knowledge from existing database which involves analyzing faulty datasets and threats to software during its lifetime functionality for building a successful intelligent software which values companies needs accordingly parameter estimation. This paper organized as: three parts for problem definition for this we begin by defining the research area at the first place followed by formulating a goal for the review process ending up into a precise statement of problem upon which future research is to be conducted. The research area is the research conducted software reliability prediction using machine learning techniques. The aim of this survey is to formulate a classification framework for the machine learning techniques used for fault prediction and help us formulate the precise problem statement. The scope of the study is the literature from 2006–2018. In part 2, we formulate the criteria for selection of literature relating to the scope of study. Several online databases were searched to provide the details for work being done in this field. These databases are: Springer, IEEE Transactions, Elsevier, Taylor &Francis, Oxford, ACM, Science Direct. The literature was searched based upon the keywords 'Software Reliability Prediction', 'fault prediction', 'failure analysis', 'machine learning', and 'artificial intelligence'. Operators were used for consideration the different possible combinations. Out of all of the total articles

which came up only those which were related to the software reliability and machine learning were taken up. A total of 40 papers were chosen for this survey.

II. Related Work

For starting research, defining research problem is the most crucial step for any research paper, based on shortcoming of existing architecture, research problem on software reliability formulated, so papers from online repository like Conference papers, Journal Papers, Workshops Tutorials, White Papers of Industry Oriented research problem relating to Preventing failure of software under specified conditions and time. Among several models, Non Homogeneous Poisson Process, papers related to attribute are taken into account: Test Driven, Continuous Integration, Coding Environment, and Feedback available from customer, SRS documents or documents available for distribution of workload, testing tools using β testing in which customer checks the software working in real environment, testing data for two point validation, interaction with customer [1][12]. So in real world scenario model based on Non-Homogenous Poisson Process proved accurate for software prediction using Stochastic Model, Wavelet Model, Topic Model, Service Architecture, Learning Algorithm, Genetic Algorithm, Reusability, Expectation Minimization, Entropy Method based on MCDN, TOPSIS, Gaussian Model, Queuing Theory, Decision Tree based on these approaches articles were chosen for classification as given in Table 1.

TABLE I. Distribution of Papers according to Approach Used

Authors	*Approach Used*
Malhotra et al. (2018)	Predictive Modelling
Sedaghatbaf et al. (2018)	Parameters Uncertainty
Choudhary et al. (2017)	Harmony Search
Singh et al. (2017)	Entropy-MCDN
Perez et al. (2017)	Stochastic Modelling
Akbar et al. (2017)	A-Z Model
Chen et al. (2017)	Topic Model
Davis et al. (2017)	Cloud Computing: IAAS
Huang et al. (2017)	Queuing Theory
Santos et al. (2017)	Markov chain Monte Carlo
Shahin et al. (2017)	Continuous Integration
Kim et al. (2017)	Reengineering
Singh et al. (2016)	Fuzzy
Tang et al. (2016)	Stochastic Model, Intelligent Software on Wavelet Model
Chen et al. (2016)	Topic Model
Machida et al. (2016)	Stochastic Modelling
Shahin et al (2016)	Service Architecture
Santos et al. (2016)	Learning Algorithm
Mende et al. (2016)	Genetic Algorithm
Peng et al. (2015)	Reusability
Lane et al. (2015)	Expectation Minimization
Wiley et al. (2015)	MCDN
McGeal et al. (2012)	TOPSIS
McAndrew et al. (2011)	Gaussian Model
Yun et al. (2009)	Queuing Theory
Zen et al. (2009)	Decision Tree

For choosing base paper, set of 40 papers based on keywords were searched and studied with following distributions of papers as in Table 2, keeping in mind sources of all the relevant papers.

TABLE II. Distribution of Research Papers according to Source

Research Papers	*Numbers*
Transactions	4
Conference	16
Workshops	5
Journals	12
White Papers	3

For specialization in research domain papers from years 2006 to 2018 studied, table was constructed based on advancement of research in this area in a chronological order based on are of work and journal in which results were published as Failure of software is very ominous representing them in terms of mathematical modeling depending on input factors. Factor Analysis [11][10][15]: As per Kaiser Criteria, Eigen values or characteristic roots are criteria for determining a factor. If Eigen values greater than one, then that is consider as a factor otherwise not a factor [9]. Exploratory Factor Analysis: It is method based on no prior theory and factors or variable can be associated with any factor. Software Reliability Models plays a important role in decision making by top executives for accessing various software reliability growth. Models based on, Non-Homogenous Poisson Process [11][21] have been proved successful and robust tool for predicting, controlling, and assessing software reliability. For the decade's research advancement in fault prediction, Software Reliability Growth Models were instigated in a impeccable debugging situation analysing any of the organization in any software fault & failure, it must be taken into care that organization operate in a way of the economy. Thus, the failure of software in the respective sector leads to organization failure, all the faults must be analysed to forecast. Forecasting the future of any company has been in economics and papers correspondingly published in Transactions, Conferences, Workshops, Journals . Figure 2 is designed keeping in mind the distribution of techniques in to that of machine learning in software reliability.

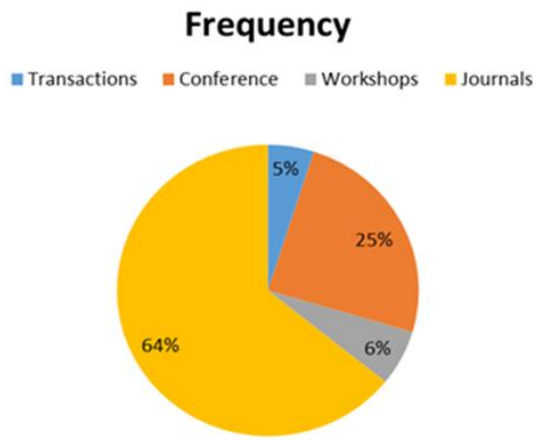

Fig. 2. Distribution of Papers source in Pie-Chart

III. Proposed Approach

In this paper, a hybrid new approach of fault prediction based on machine learning algorithm is proposed by taking a survey of papers from 2006-2018 on machine learning algorithms. Figure 3 depicts a framework based on software reliability and machine learning algorithms.

Algorithms for Classifying Faulty Datasets

Learning Algorithm: These learning techniques are based on prior experience of developers on handling such the datasets

Decision Tree: It is learning by predefined classes on data is categorized by tree data structure where each node test is assigned, branch on outcome result and leaves for classification of dataset accordingly. So, it a way of supervised learning for classifying new dataset based on previous classes classified [22][24][20].

Clustering: Learning by defining clusters on datasets according to Partition, Hierarchical based on top-down, bottom-up dendograms as Agglomerative & Divisive, DBScanbased on the nearest distance of clusters and number of clusters specification is not needed previously as minimum distance calculated by minimum, maximum, average, Sum of Squared Errors (SSE) based on these values clusters are merged together and updated in Matrix based on numerical value between original and new clusters till single cluster formed[13][6][4].

Fuzzy Set Theory is based on dataset defined faults caused by them may lead to software failure To determine intensity of failure on Non- between $(0 \ldots \ldots 1)$.Defects are assigned to the priority basis as some defects removal is very important in causing failures. Membership Function $F(x) = \frac{1}{1+\left|\frac{x-w_2}{w_0}\right|^{2w_i}}$

Neurofuzzy: It is a hybrid approach based on neural networks and fuzzy logic, can handle computational problems smartly by providing optimal solution on incomplete, partisan datasets by self-organizing based on learning by itself without need of any prior knowledge of handling such datasets [3][11]

Evolutionary algorithms: The algorithmic techniques provoked by biological evolution such as reproduction, selection, mutation and recombination [15].

Differential Evolution (DE): Algorithmic Techniques based on stochastic approach of handling and solving optimization problems by mutation and crossover for software defect handling at real time

Revised Mutation Strategy: It tries to find best solution vector $Y_{bs,G}$providing early solution as compared with random vector generation it considers two variables P_{old} for old population and P_{new} for new population produced during mutation

Let F= Mutation Factor in interval $(0,2)$

Y' =Optimal Value Produced

N = Varying Factor in interval$(0,1)$,

$$Y' = Y_{bs,G} + N.\left(Y_{bs,G} - P_{old}\right) - F.Y_{bs,G}$$

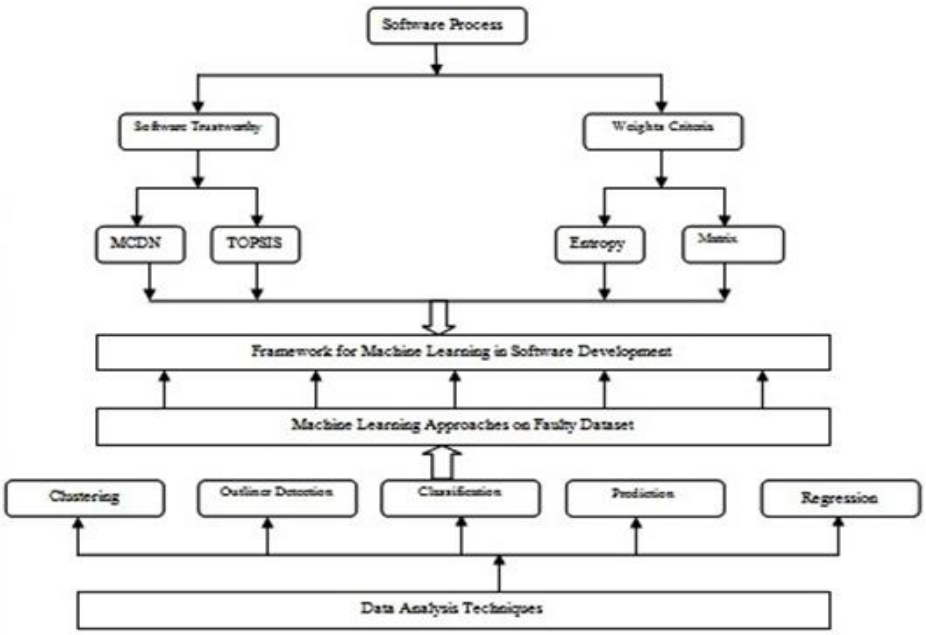

Fig. 3. Proposed Framework of Analyzing software reliability prediction using Machine Learning Algorithm

Modeling before Software Release

Assuming:

τ_{i-1} =Time for ith release ($i = 1\ to\ 4$)

a_i = Initial Fault Content for ith release ($i = 1\ to\ 4$)

a = Total Initial Fault

$\otimes$ =Stieltjes Convolution

$*$= Convolution Operator

b_i = Fault detection rate for ith release ($i = 1\ to\ 4$)

β =Constant

$a = ExpectedNumberoffailure,$
$b = Faults\ Detection\ Rate$

$$m(t) = a(1 - ((1 + bt)e^{-bt}), where\ a > 0, b > 0$$

$$m(t) = a(1 - e^{-bt}), where\ a > 0, b > 0$$

$$m(t) = N\left((1 - \frac{\beta}{\beta + (c/b)\ln((a + e^{bt})/(1 + a))})\right)^{a}$$

Reliability Prediction using Discrete Fourier Transform:

In this modeling$1/N$ parameter factor is calculated based on the value of repeat occurrence of faults or frequency for faults, from previous dataset its occurrence could be validated by I[25][18][16].

p_i =Proability of perfect release for ith release ($i = 1\ to\ 4$)

α_i =Error Generation Rate for ith release ($i = 1\ to\ 4$)

s =Testing Time

u =Resources Allocated

$m(s,u)$ =Cumulative number of faults removed by time s with usage of resources u.

p =Proability of perfect debugging.

α =Error Generation Rate.

Probability of Faults Rate $P(F)$ =Constant, whether fault produced or not during correction process

Expected $E = [m_i(\tau)]$ =Number of faults removed by time τ for software release

$f(\tau), g(\tau) =$ Probability cumulative density function $F(\tau)$ =Probability Distribution for Fault Detection.

$G(\tau)$ =Probability Distribution for Fault Correction.

$$F_{X<Y}(t) = Pr_{X<Y}(T \le t)$$
$$= \int_{0}^{t} \int_{\frac{\gamma_Y}{\gamma_{Y_*}}\tau_1}^{t-\tau_1+\frac{\gamma_Y}{\gamma_{Y_*}}\tau_1} \left(\gamma_{Y} e^{-\gamma_Y \tau_1}\right)\left(\gamma_{Y_*} e^{-\gamma_Y \tau_2}\right) d\tau_2\, d\tau_1$$

$$y_i^{-I} = f\left(\sum_{j=1}^{N_{I-1}} w_{ij}'.\bar{y}^{I-1} + \theta_i^{I}\right),$$
$$= 1, \dots, N_I, l = 1, \dots, L$$

Model Parameter Estimation:

Based on Stochastic Modeling, parameter estimation technique Non- Homogenous Poisson Distribution

Modeling Assumptions:

1. Using SRGM, detection and removal of faults are modeled.
2. Fatal loss due to failure of software system as faults remained present due to debugging.
3. Detecting and correcting faults be in suitable time lag.
4. Faults remaining in the software due to fault correction some new faults generate lead to failure, thereby affects software reliability.
5. If we are decreasing faults before software release, then Fault rate is decreased by 1 with the probability p or Fault rate remains same with probability $1 - p$.

IV. Discussion

For the decade's research advancement in fault prediction, Software Reliability Growth Models were instigated in a impeccable debugging situation but debugging remained faulty and erroneous, for faulty debugging models devised by. using Genetic Algorithm [11][6][9]. The model proposed on Topic Model [5][7][9] model for finding faults using exponential S curve. This model is referred as imperfect debugging model. Models were suggested on a structure arranged on Decision Tree [8][13] for considering imperfect debugging phenomenon to develop testing effort-dependent fault detection and fault correction process models. A hybrid general framework for deriving several software reliability growth models based on Fuzzy in the occurrence of in accuracy group and fault fixing. [14][5] Models with Stochastic Model function in faulty debugging environment with constant and time-varying fault detection rates [2] developed a two-stage fault detection and correction model under the effect of two types of imperfect debugging for Entropy [MCDN] multiple releases of the software. [8][12] proposed a multi-up gradation model in Cloud imperfect debugging environment. [7][9] considered a log logistic distribution fault content function in an imperfect debugging model.[5] presented a two-stage detection- correction-based model with queuing theory testing effort incorporating imperfect debugging in multi up gradation of software. [9][1] They assumed exponential distribution as detection Continuous Integration.[3] Classification process and logistic distribution function as correction process. Models for fault prediction based on hybrid EM Algorithm, Rough Set Theory Bayesian Theory [4][7][15], K-Mean-Clustering Reengineering Hybrid: Decision Tree with Bayesian Fuzzy &Apriori [7] [10].

V. Conclusion and Future scope

Based on review of papers from 2006-2018, software reliability techniques studied and based on time frame techniques changing from time to time, so papers are classified according to Author's techniques, Yearwise distribution, Approaches distribution. After reviewing future work for designing compressive approach for intelligent software systems Modeling designed before releasing software based on Machine Learning Techniques.

REFERENCES

[1] An empirical analysis of the effectiveness of software metrics and fault prediction model for identifying faulty classes, Computer Standards & Interfaces, v.53, pp.1-32, August 2017

[2] Bavota, G., Gethers, M., Oliveto, R., Poshyvanyk, D., & Lucia, A. D. (2014). Improving software modularization via automated analysis of latent topics and dependencies. ACM Transactions on Software Engineering and Methodology (TOSEM), 23(1), 4.

[3] Chen, T. H., Thomas, S. W., Nagappan, M., & Hassan, A. E. (2012, June). Explaining software defects using topic models. In Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on (pp. 189-198). IEEE.

[4] Chen, T. H., Thomas, S. W., Nagappan, M., & Hassan, A. E. (2012, June). Explaining software defects using topic models. In Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on (pp. 189-198). IEEE.

[5] Daniel Rodriguez, Israel Herraiz , Rachel Harrison , Javier Dolado , José C. Riquelme, Preliminary comparison of techniques for dealing with imbalance in software defect prediction, Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, May 13-14, 2014, London, England, United Kingdom

[6] Huang, C. Y., &Kuo, T. Y. (2017). Queueing-Theory-Based Models for Software Reliability Analysis and Management. IEEE Transactions on Emerging Topics in Computing, 5(4), 540-550.

[7] K. Muthukumaran , N. L. Bhanu Murthy , G. Karthik Reddy , M. Aruna, Comparative study on effectiveness of standard bug prediction approaches, Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop, p.1-4, October 17-19, 2013, New Delhi, India

[8] Kim Herzig, NachiappanNagappan, Empirically detecting false test alarms using association rules, Proceedings of the 37th International Conference on Software Engineering, May 16-24, 2015, Florence, Italy

[9] M., Treleaven, P. and Yingsaeree, C. (2011) 'Algorithmic trading', IEEEComputer Society, Vol. 44, No. 11, pp.61–69

[10] Machida, F., Xiang, J., Tadano, K., &Maeno, Y. (2017). Lifetime extension of software execution subject to aging. IEEE Transactions on Reliability, 66(1), 123-134.

[11] Mende, T., &Koschke, R. (2010, March). Effort-aware defect prediction models. In Software Maintenance and Reengineering (CSMR), 2010 14th European Conference on(pp. 107-116). IEEE.

[12] Meng Yan , Yicheng Fang , David Lo , Xin Xia , Xiaohong Zhang, File-level defect prediction: unsupervised vs. supervised models, Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, November 09-10, 2017, Markham, Ontario, Canada

[13] Nuti, G., Mirghaemi, YuchenGuo , Martin Shepperd , Ning Li, Bridging effort-aware prediction and strong classification: a just-in-time software defect prediction study, Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings, May 27-June 03, 2018, Gothenburg, Sweden

[14] Peng, K. L., & Huang, C. Y. (2017). Reliability analysis of on-demand service-based software systems considering failure dependencies. IEEE Transactions on Services Computing, 10(3), 423-435.

[15] Rumbaugh, J., Jacobson, I., &Booch, G. (2004). Unified modeling language reference manual, the. Pearson Higher Education.