Modeling Attributes Influencing Intelligent Information Software System using Exploratory Factor Analysis

Manu Banga, Abhay Bansal, Archana Singh

Abstract*: *In today's era Software development is hybrid with Software Reliability Model, using model parameters for validating software faulty dataset as it has many pitfalls choosing reliability model being dataset biasing, parameters on which choosing parameter estimation relevant to model based on statistical test, readiness index, functionality preserving, effort consumed on faulty datasets prediction. Model Estimation Modeling based on Least Square Error (LSE) for which factor analyzed using Exploratory Factor Analysis and Confirmatory Factor Analysis and model is validated using p-test with confidence interval of 95% on cloud service. Based on our eleven attribute of our dataset, Continuous Integration is most relevant among other factors. Defect Prediction is one the most crucial and critical task in successful operation of software working as can leads to faults which further causes failures and these failures are very ominous, thus reducing and preventing software defects is major challenging work

Keywords—Software Reliability Models, Cloud Service, EFA, CFA, Intelligent Information System.

I. INTRODUCTION

Software Reliability Growth Models in software development life cycle of debugging failures intensity using testing history for predicting behavior of software in real time environment. Firstly, selection of model, which has eigen-values calculated in a performance matrix, lastly parameters are estimated using LSE based on EFA and CFA. Collectively, these are four steps are known as Model Composition[14,15]

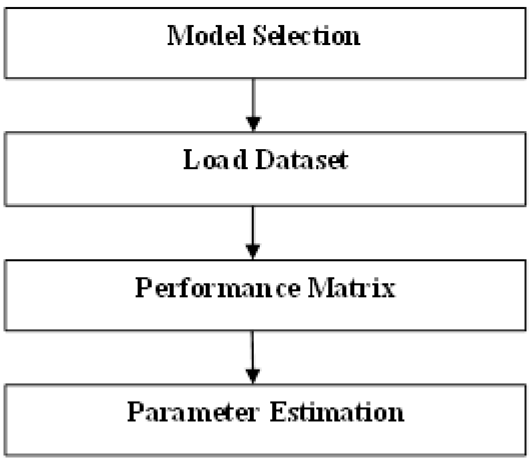

Figure 1: Model Composition

Among all the software reliability models Non-Homogenous Poisson process based on S-shaped, concave curve software-debugging phenomenon.Based on software failure phenomenon causes monetary losses, time loss, as faults transform to failures [1, 2]. Thus controlling faults at appropriately and careful examine them in a testing/debugging phase by using previous data of software failure and 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 is accessed. Among all the available models, Non-Homogenous Poisson Process Model has been employed to estimating &predicting remaining number of defects in a software system under test [4, 9]. Software Reliability Models plays a important role in decision making by top executives for accessing various software reliability growth. models based on practical software reliability engineering, NHPP SRGMs have been proved successful and robust tool for predicting, controlling, and assessing software reliability.

II. LITERATURE REVIEW:

Preventing failure of software under specified conditions and time. Among several models, Non Homogeneous Poisson Process, 11 attribute are taken into account: Test Driven, Continuous Integration, Coding Environment, 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 2 point validation, interaction with customer [7, 8]. So in real world scenario model based on Non-Homogenous Poisson Process proved accurate for software reliability prediction [5, 6].

Attribute	Description
T. D(Test Driven)	Code Refactoring
C.I(Continuous Integration)	Integration
Coding Environment	Defects Detection
Feedback	Iteratively Detection at each stage
Documents	For Distribution of workload
Testing Tools	βeta Testing,
Testing Data	For Validation Purpose
Interaction	Periodic Meeting

Revised Manuscript Received on June 14, 2019
Manu Banga, ASET, Amity University, Noida, India.
Abhay Bansal, ASET, Amity University, Noida, India.
Archana Singh, ASET, Amity University, Noida, India.

Test Cases	Checking Functionality
Testing Method	Iterative Check
Real Environment	At real time

Table 1: Attribute affecting testing of Model

Failure of software is very ominous so in terms of mathematical modeling depending on input factors. **Factor Analysis:** As per Kaiser Criteria, Eigen values or characteristic roots are criteria for determining a factor. If eigenvalues greater than one then that is consider as a factor otherwise not a factor [3].

Exploratory Factor Analysis: It is method based on no prior theory and factors or variable can be associated with any factor [1].

Confirmatory Factor Analysis: It is a method based on pre established theory, assuming each Factor associated with specified sets of measured variables. It is used of evaluating evidences [2]. .

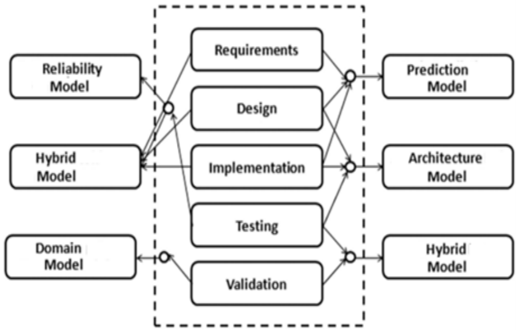

Figure 1: Reliability Models on software development phase.

Model Parameter Estimation:

Based on Stochastic Modeling, parameter estimation technique

$$a = ExpectedNumberoffailure,$$

$$b = Faults\ Detection\ Rate$$

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

represented concave curve,

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

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

Non- Homogenous Poisson Distribution:-

Laplace Trend Test: By this software reliability growth is predicted. It is based on centroid and after its comparison scope is calculated and if its secure is greater than zero than it is a upward slope and if its secure is less than zero then it is a downward slope and slope of zero means horizontal line [12].

$$LTT(x) = \frac{\sum_{k=1}^{x}(k-1)u(i) - \frac{x-1}{2}\sum_{k=1}^{x}u(i)}{\sqrt{\frac{k^2-1}{12}\sum_{k}^{x}u(i)}}$$

where$u(i)$ represents number of failures in a time t, positive value of function $LTT(x)$ predicts decrease in reliability and negative value predicts increase in reliability [12].

Mean Square Error = $\frac{\sum_{x=1}^{i}(m(t_x)' - (m(t_x)^2}{i-p}$,

Signal Noise Ration ($R_i = -10\log(\sum_{x_{n=1}}^{N_{x_n}} \frac{y_{x_{n=1}}}{N_{x_n}})$,

where,

$$R_i = numberofexpeiments\ trails,$$

$$x_n = Number\ of\ times\ trail\ sequence,$$

$$N_{x_n} = RandomTimesoccuredinsequence$$

$$y_{x_n} = Number\ of\ generations\ required\ for\ require\ results.$$

1. **Generalised Goel NHPP Model:**

Mean Value Function:

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

Software Failure Intensity $\lambda(t) = abct^{c-1}e^{-bt^c}$

2. **Goel Okumoto Model:**

Mean Value Function:$m(t) = a(1 - e^{-bt})a > 0, b > 0$

Software Failure Intensity$\lambda(t) = abe^{-bt}$

3. **Gompertz Model:**

Mean Value Function: $m(t) = ake^{-bt}$

Software Failure Intensity$\lambda(t) = ab\ln(k)k^{e^{-bt}}e^{-bt}$

4. **Inflection S-Shaped Model:**

Mean Value of function $m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}}$

Software Failure Intensity$\lambda(t) = \frac{ab\, e^{-bt}(1+\beta t)}{(1+\beta e^{-bt})^2}$

5. **Logistic Growth Curve Model:**

Mean Value Function:

$$m(t) = \frac{a}{1 + ke^{-bt}} a > 0, b > 0, k > 0$$

Software Failure Intensity:$\lambda(t) = \frac{abke^{-bt}}{(1+ke^{-bt})^2}$

6. **Modified Duane Model:-**

Mean Value Function

$$m(t) = a[1 - (b/(b + t))^c]a > 0, b > 0, c > 0$$

Software Failure Intensity:$\lambda(t) = acb^c(b + t)^{1-c}$

7. **Musa-Okumoto Model:**

Mean Value Function $m(t) = \alpha\ln(1 + \beta t)$

Software Failure Intensity$\lambda(t) = ab/(1 + bt)$

8 **Yamada Imperfect Debugging Model:**

Mean Value of function

$$m(t) = \frac{ab\left(e^{at} - e^{-bt}\right)}{a + b}$$

Software Failure Intensity

9 Yamada Rayleigh Model:

Mean Value of function

$$\boldsymbol{m(t)} = \boldsymbol{a}\left(\boldsymbol{1 - e}^{-r\alpha\left(\left(1-e^{\left(-\beta t^2/_2\right)}\right)\right)}\right)$$

$$\boldsymbol{a} > 0, r\alpha > 0, \beta > 0$$

Software Failure Intensity

$$\boldsymbol{\lambda(t) = ar\alpha\beta te}^{-r\alpha\left(\left(1-e^{\left(-\beta t^2/_2\right)}\right)\right)}$$

10 Delayed S- Shaped Model:

Mean Value of function$\boldsymbol{m(t) = a\left(1 - (1 + bt)e^{-bt}\right)}$

Software Failure Intensity$\boldsymbol{\lambda(t) = ab^2e^{-bt}}$

11 Yamada Imperfect Debugging Model 2:

Mean Value of function

$$\boldsymbol{m(t) = a\left(1 - e^{-bt}\right)\left(1 - \frac{\alpha}{b}\right) + \alpha at}$$

Software Failure Intensity

$$\boldsymbol{\lambda(t) = abe^{-bt}\left(1 - \frac{\alpha}{b}\right) + \alpha a}$$

12 Yamada Exponential:

Mean Value of function

$$\boldsymbol{m(t) = a\left(1 - e^{-r\alpha\left(1-e^{-\beta t}\right)}\right)}$$

Software Failure Intensity

$$\boldsymbol{\lambda(t) = ar\alpha\beta e^{-r\alpha\left(1-e^{-\beta t}\right)-\beta t}}$$

13 Pham-Nordmann-Zhang Model (PNZ model):

Mean Value of function

$$\boldsymbol{m(t) = \frac{a\left(1 - e^{-bt}\right)\left(1 - \frac{a}{b}\right) + at}{1 + \beta e^{-bt}}}$$

Software Failure Intensity

$$\boldsymbol{\lambda(t) = \frac{abe^{-bt}\left(1 - \frac{a}{b}\right) + a}{1 + \beta e^{-bt}} + \frac{ab\beta e^{-bt}\left(1 - \frac{a}{b}\right) + a}{1 + \beta e^{-bt^2}}}$$

14 Pham-Zhang NHPP (PZ Model):

Mean Value of function

$$\boldsymbol{m(t) = \frac{1}{\left(1 + \beta e^{-bt}\right)}\left((c + a)\left(1 + e^{-bt}\right) - \frac{ab}{b - a}\left(e^{-at} - e^{-bt}\right)\right)}$$

Software Failure Intensity:

$$\boldsymbol{\lambda(t) = \frac{b(c + a)(1 + \beta)e^{-bt} - \left[be^{-bt}\left(1 + \beta e^{-bt}\right) - ae^{-at}\left(1 + \beta e^{-bt}\right)\right]}{\left(1 + \beta e^{-bt}\right)}}$$

15 Pham-Zhang Imperfect Fault Detection (Pham-Zhang IFD model):

Mean Value of function

$$\boldsymbol{m(t) = \alpha - \alpha e^{-\beta t}\left(1 + (\beta + \gamma)\tau + \beta\, d\tau^2\right)}$$

Software Failure Intensity:

$$\boldsymbol{\lambda(t) = \alpha e^{-\beta t}\left(1 + (\beta + \gamma)\tau + \beta\, d\tau^2\right)}$$

16 Zhang-Teng-Pham Model:

Mean Value of function

$$\boldsymbol{m(t) = \frac{\alpha}{\rho - \beta}\left[\left(1 - \frac{\left(1 + \alpha e^{-b\tau}\right)}{1 + \alpha e^{-b\tau}}\right)^{\frac{c}{b}(\rho-\beta)}\right]}$$

Software Failure Intensity

$$\boldsymbol{\lambda(t) = \frac{\alpha c}{1 + e^{-b\tau}}\left[\left(\frac{1 + \alpha e^{-b\tau}}{1 + \alpha e^{-b\tau}}\right)^{\frac{c}{b}(\rho-\beta)}\right]}$$

III. PROPOSED APPROACH:

In this paper, a new algorithm is proposed in which we are doing factor analysis on 11 attribute on standard datasets available over cloud relevant. Based on estimation model, Matrix was made on the variance vales, standard deviation was calculated of the dataset.

Membership Function $F(x) = \frac{1}{1+\left|\frac{x-w_2}{w_0}\right|^{2w_i}}$

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 are very as important in causing failures.

Reliability of Cloud Based Modeling: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

$$\boldsymbol{F_{X<Y}(t) = Pr_{X<Y}(T \le t)}$$

$$= \int_{0}^{t} \int_{\frac{\gamma_Y}{\gamma_{Y*}}\tau_1}^{t-\tau_1+\frac{\gamma_Y}{\gamma_{Y*}}\tau_1} \boldsymbol{\left(\gamma_{Y^{e^{-\gamma_Y\tau_1}}}\right)\left(\gamma_{Y_*{}^{e^{-\gamma_Y\tau_2}}}\right) d\tau_2\, d\tau_1}$$

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

$$i = 1, \ldots, N_I, l = 1, \ldots, L$$

Homogenous Poisson Process Model we apply our novel approach:

1. Preprocessing the Dataset using Factor Analysis based on faulty dataset.
2. Applying Exploratory Factor Analysis on the faulty dataset, parameters are investigated
3. After step 2, Confirmatory Factor Analysis was applied on the dataset, validating factors relevant to estimating parameter relevant to our model

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

4. New Variance, σ_i again calculate of the dataset and σ_i compared with σ_{i+1}, based on this Matrix is based.

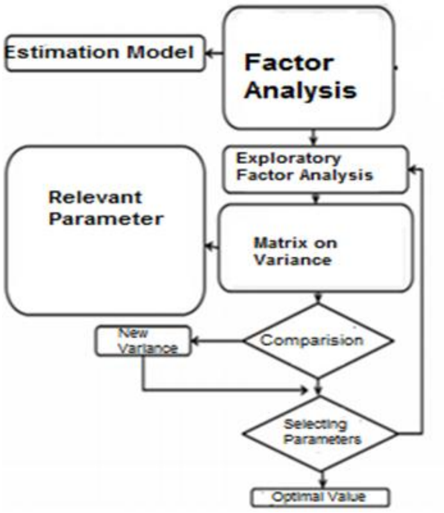

Figure 2: Modeling Attribute using Exploratory Factor Analysis

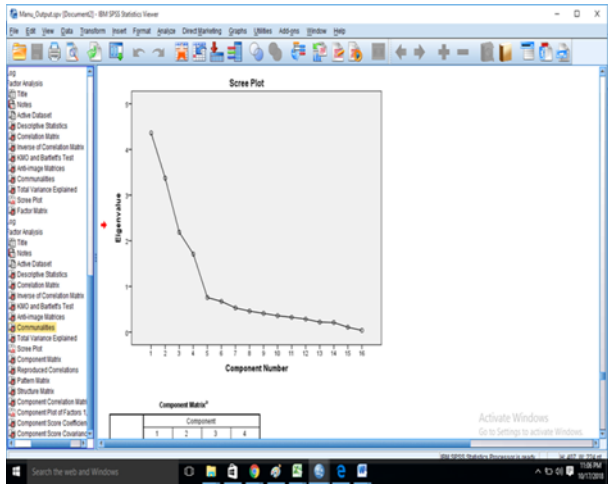

Figure3 : Effect of change of Eigen Values on Variance

Table 2: Attribute with Eigen Values

Attribute	Eigen-values Coefficient	% of Variance	Total	Attribute	Eigen-values Coefficient	% of Variance	Total
Test Driven Development	4.348	25.175	25.175	Requirement	0.414	2.585	89.16
Continuous Integration	3.365	21.034	45.209	Test Cases	0.366	2.288	91.448
Test Environment	2.391	11.945	60.154	Method Used	0.334	2.087	93.535
Feedback	1.41	11.811	71.965	Customer	0.284	1.776	95.311
Documentation	0.688	3.301	75.266	Defects	0.278	1.735	97.045
Tools	0.635	3.969	80.235	Realistic	0.215	1.343	98.388
Data	0.547	3.421	83.656	Testing	0.167	1.043	99.431
Meetings	0.467	2.919	86.565	Test Driven Development	0.091	0.569	100

IV. CONCLUSION:

In this paper, a new technical analysis was done on Exploratory Factor Analysis and then applying Confirmatory Factor Analysis for Model Estimation mathematical analysis based on factor selection relevant among the given eleven factors based on Readiness Index values, Parameter Dependency,

REFERENCES:

1. B. O ztay¸si, T. Kaya, and C. Kahraman, "Performance comparison based on customer relationship management using analytic network process," Expert Systems with Applications, vol. 38, no. 8, pp. 9788–9798, 2011.
2. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–7
3. G. Adomavicius and Z. Jingjing, "Stability of Recommendation Algorithms," ACM Transactions on Internet Technology, vol. 10, no. 4, 2012.
4. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Transactions on Information Systems, no. 1, pp. 5– 53, 2004.
5. J.-Y. Jiang, S.C. Tsai, and S.J. Lee, "FSKNN: multi-label text categorization based on fuzzy similarity and K nearest neighbors," Expert Systems With Applications, vol. 39, no. 3, pp. 2813–2821, 2012.
6. M. Balabanovi and Y. Shoham, "Fab: content-Based, collaborative recommendation," Communications of the ACM, vol. 40, no. 3, pp. 66–72, 1997.
7. M. J. Pazzani, "A framework for collaborative, content-based, and demographic filtering," Artificial Intelligence Review, vol. 13, no. 5-6, pp. 393–408, 1999.
8. M. L. Zhang and Z. H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning," Pattern Recognition, vol. 40, no. 7, pp. 2038–2048, 2007
9. R. Liu, J.G.Liu, C.-X.Jia, D. Sun, and B.-H.Wang, "Personal recommendation via unequal resource allocation on bipartite networks," Physica A, vol. 389, no. 16, pp. 3282–3289, 2010.
10. S. Maslov and Y.-C.Zhang, "Extracting hidden information from knowledge networks," Physical Review Letters, vol. 87, no. 24, Article ID 248701, 2001.
11. S. Singh, B. P. S. Murthi, and E. Steffes, "Developing a measure of risk adjusted revenue (RAR) in credit cards market: implications for customer relationship management," European Journal of Operational Research-, vol. 224, no. 2, pp. 425–434, 2013.
12. T. Zhou, L.-L.Jiang, R.-Q.Su, and Y.-C. Zhang, "Effect of initial configuration on network based recommendation," Europhysics Letters, vol. 81, no. 5, Article ID 58004, 2008.
13. Y. M. Yang, "An evaluation of statistical approaches to text categorization," Information Retrieval, vol. 1, no. 1-2, pp. 69–90,1999.
14. Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, and F. Yang, "Recommendation model based on opinion diffusion," Europhysics Letters, vol. 80, no. 6, Article ID 68003, 2007.
15. Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," ACM Transactions on Information Systems, vol. 22, no. 1, pp. 116–142, 2004.

AUTHORS PROFILE

Manu Banga, Ph.D. Scholar (CSE), ASET, Amity University Noida. He has done M. Tech and B.Tech and published various papers in conference and journals.

Dr. Abhay Bansal, Jt. Head ASET / Director, DICET, HOI. ASET, Amity University, Noida, His area of research is data mining, big data, and software engineering. He guided more than 15 Ph.D. scholar He is life member of ACM, IEEE, CSI & IET.

Dr. Archana Singh, Ph.D. (CSE), M.Tech(CSE). She is working as Professor at ASET, Amity University, Noida Her area of research is data mining, big data, software engineering. Presently she is Ph.D. guide of 8 students She is life member of ACM, IEEE, CSI & IET