Multidisciplinary Review Journal

Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 2, 2020

Review Article

STATISTICAL QUARTILE DEVIATION-BASED SOFTWARE RELIABILITY GROWTH ESTIMATION MEASURE FOR RELIABILITY PREDICTION

Y. Geetha Reddy[1] , Dr. Y Prasanth[2]

Department of Computer science and Engineering
Koneru Lakshmaiah Education Foundation, Guntur District, A.P., India.

Received:16.11.2019 ***Revised: 18.12.2019*** ***Accepted: 20.01.2020***

Abstract

Software reliability growth models are used to predict the quality of the software systems using statistical learning models. However,a large number of faults areremained undetected in small and medium applications. A large number of traditional reliability measures are used to test the software faults in theapplication development and testing process. But in real-time, new faults are included in the software testing and maintenance phases in order to find the reliability estimation. The main problem in the existing SRGMs include, difficult to handle large reliability data and these models are not applicable to statistical dependencies and independency measures. In the proposed model, a novel statistical dependency and independency-based quartile density distribution model is implemented to improve the reliability prediction rate. Experimental results proved that the present model has high reliability estimation probability than the traditional growth models in terms of skewness and peak-ness.

Keywords: Software reliability estimation, statistical growth models, software faults.

© 2019 by Advance Scientific Research. This is an open-access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)
DOI: http://dx.doi.org/10.31838/jcr.07.02.64

INTRODUCTION

The majority of computer reliability growth models describe the software testing cycle as time. Numerous factors, such as the test method and the environment, often affect test failure. Data performance is one of the key characteristics of software quality and is the main quantitative criterion for measuring software quality, which is why researchers are paying more and more attention. Computer reliability modeling is one of the main areas of computer reliability theoretical science and engineering practice. This seeks to assess the state and behavior of system reliability, to help develop reliable technology and to test software reliability. To date, nearly 100 types of software reliability models such as the G-O models[1], the M-O models[2] and the J-M models[3] have been released by researchers. Nonetheless, these models are basically non-linear function models; their parameters are difficult to estimate.

Because the computer consistency method isn't in linear, a new idea is to use the smart optimization algorithm for system parameter estimation. A group intelligent optimizing algorithm, the artificial bee colony algorithm (ABC), is characterized by simple operation, little check parameters and excellent performance. It was commonly used in the fields of task optimization, pattern recognition, moving target tracking and robot route planning.

SRGMs are mathematical models that systematically evaluate the performance of data to ensure the quality of software. SRGM quantifies the software's performance by creating the numerical relation between the testing time and the frequency of failure during the testing procedure. Such relationships are defined in terms of stochastic processes, probability and statistical formulas. The first step involves estimating the cumulative error numbers stochastically over a period of time. The mean value function is defined as the error detection rate of' a(t)' and' b(t)' in the next step. The model is eventually evaluated and the mean value function determined using test data. MVF is a calculation of the amount of cumulative computer shortcomings at any given time. Computer performance has traditionally been measured mainly from test data. The code was usually tested when most of its functions were developed and its usage profile used to equate test data with system performance, as required by its application.It involves a comparative performance, predictability, and precision study of SRGM models based on thirty-eight failure data sets that cover system error data, field data and OSS fault information.

In the performance test SRM is fitted with data collected with statistical techniques (e.g. Linear Regression, Nonlinear Regression) based on the type of data collected. The maximum expected future losses was estimated on the basis of a performance forecast fitted with an SRM. All forecasts and predictions require good data, i.e. data are correctly reported when failures occur and appropriate, i.e. data are related to the prediction's specific environment[9]. Software systems in an environment similar to the operating environment were tested for performance testing.

The first growth model of computer quality was introduced in 1972[4]. In order to overcome the unstable J&M model problem, researchers used the maximum likelihood method to estimate parameters. The Yamada et al. proposed that and analyzed the error detection system in 1983 and found that the number of errors observed in the analyzed data were S-shaped, so the growth model was called s-shaped[5]. A structural approach for the identification of software residual defects was published in[6] et al. The researchers believed that errors using this method could lead to discovered errors being detected without software failure. Yamada et al. proposed a growth model focused on the necessary research effort for the study period[7]. Weibull Curve was used to connect the research efforts at that time. NHPP[8] was the template technique. There have also been suggestions for many other parametric models. Most models of computer reliability include underlying assumptions such as time interval dependency, immediate corrections of the error found and the implementation of no new defects during the fault correction. Such expectations seem unrealistic[9]. One of the key drawbacks of the parametric SRGMs is that computer behavior changes as software codes change during the process. In real software development, the theoretical adaptability of stochastic methods will usually not be sufficient to the assumptions addressed in parametric SRGMs. Another important issue is that no such template can be used to accurately predict different software projects[10]. Good researchers used non-parametric SRGMS (NPSRGM) to address these limitations. Machine learning methods such as neural nets, genetic algorithms, vector support and so forth

are typically used by NPSRGMs. The advantage of these models over parametric models is that these models are not necessary and are made more accurate by the machine self-learning behaviour.Consequently, computer performance data can not be given in due time in order to make cost-effective changes to improve its own or system layout reliability. A code default is a loss of the necessary feature that may occur in different ways (failure modes). Computer faults are primarily related to design because of a design error (fault), which can transform into a fault mode because of certain conditions and/or the triggering event. That is why most or all software failures are structural in nature and their prevention results in increased software performance. The early design of software for the detection and mitigation of software faults and for the reliability or accessibility of software via fault-protection and correction of design-based errors is essential both for failure modes and effects analysis (FMEA) and functional testing. The performance of hardware was designed and calculated using the Weibull Intensity Function mathematical model of power control. The Weibull feature fits the trustworthiness growth curve. The template is well suited for largely independent failures but due to a development flaw and mitigated.

RELATED WORKS

Several research works have been proposed and validated many NHPP-based SRGMs based on hardware performance research patterns. SRGM defines the occurrence and/or the removal of failures by time (CPU time, calendar time, time execution, test case) and/or by resources used during the testing and operational phases for testing and debug software development. Some computer quality growth models differentiate the software testing process. Many variables, such as the test method and the environment, however, influence test failure. Such variables appear as shifts with clearly distinct curves before and after a change in the identification of a fault. The whale optimization algorithm(WOA) is easy to use and has little adaptation parameter that is better than the PSO, Gray Wolf optimizer (GWO), GSA, and so on. WOA is basically free from limiting assumptions about continuity, derivatives, unimodality, etc. The failure information is known as an input to the SRGM, which provides a reliability estimate as an output for mathematical functions like exponential and logarithmic functions. The first step is to measure the total number of errors over a' t' period. The mean value function will be defined as the error detection functions of a(t) and b(t) in the next step. The model is eventually evaluated using test data and the mean value function is determined. MVF is the calculation of software defects accumulated at any time ' t.' Historically, software quality has been measured primarily by testing data. Usually software has been tested to correlate test data to software reliability if most or all of their functions have been developed and their application profile used.

Several examples of the NHPP models include the model of Goel-Okumoto (GO), a NHPP with an exponential rate decay function that considers failure detection, a Gompertz Curve Growth Model (S-shaped), a model which was used by many Japanese machine fabrications and software companies.

Several SRGMs have been suggested or produced. Many of them have their own limits, theories and unique features. The model has good results for a certain data set, but no model is sufficiently good for all data sets from various domains[5]. The problem of generalization of SRGM further complicates reliability prediction model choice.

Following efforts of Crowet al.[11] in model choice, further research and studies have been carried out in order to achieve the optimal approach to model selection[12-14]. Such studies are however focused on numerical methods such as the LES and nonlinear regression (NLR) as methods of measurement of the SRGM parameters, which can be strengthened by computational intelligence (CI) methods like the PSO.

PSO is an innovative calculation and gradient-based global optimization technique that imitates the action and intelligence of fish and birds in motion. It initiates a random particle population that explores the search space for the solution. Each particle is represented by vector string co-ordinates and a random frequency. -iteration has been modified according to the rate at which the particle has reached the best fitness (pbest) and the neighborhood where they reached the best fitness in the neighborhood (gbest), and updated the particle's location within the trouble space.

Garget al.[15] studied how several computer reliability growth models were built in which the failure detection mechanism relies on residual fault and test time. In addition, they saw how these models can be inferred by using a delay effect factor as the delayed fault detection model. They have proposed 4 new SRGMs depending on the power function of the test time concept, assuming that the software has two types of faults.

Such defects were the leading defects and reliant defects. Leading defects can be eradicated immediately following a malfunction. The dependent defects are obscured by leading defects and, thus, their associated principal defects must be eliminated first with a debugging time lag. Specific code error data were used to check these models for their health, validity and applicability.

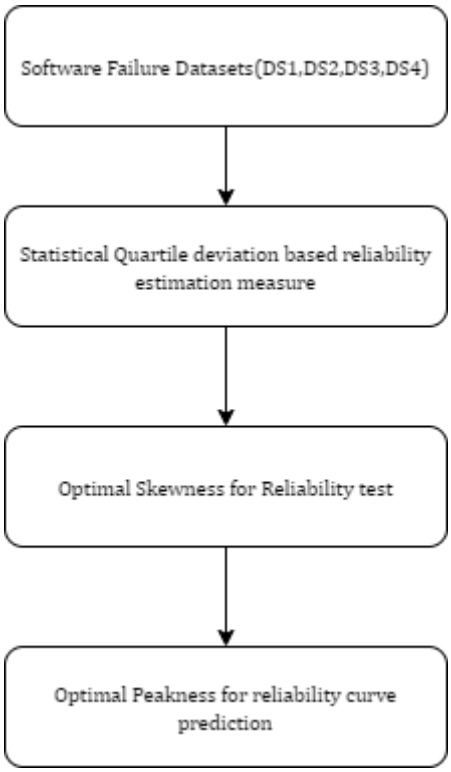
Patiet al.[16] explained how imprecise Bayesian approaches can be combined with inferior likelihood, and this is addressed using growth models of reliability. The central theme of this approach is to divide a set of relevant model parameters into two subsets that are linked to various basic features of the overall model and then combine the Walley notion of inexact Bayesian models associated with one of the sub-sets of model parameters with a maximum likelihood estimate of the remaining subset. The Bayesian model has been constructed using the first subset and statistical information, which then provides a lower and higher predictive probability distribution based on the second set of parameters. These additional parameters are calculated using a most probable process, which relies on the new proposal for a total approximation of the likelihood of sets of distributions based on imprecise Bayesian models for the other subset of parameters.

This hybrid approach is used to show the accuracy of growth models and regression models along with the discussion of subjects appropriate for validating the system and promoting it.The management's decision on the release time of the program dictates quality growth factors versus testing costs. The control of fault variations in the reliability and testing force curve of the growth curve requires controlled settings to be applied to the entire software development process. In order to integrate efficiently many of the software's system performance design models under the Non-Homogenous Poisson Process (NHPP), they have been used extensively to model error detection and corrections, which represent a test process in terms of test effort modeling, learning effects modelling, exponential or S-form fault detection frequency modeling or even change-point modeling. Some of them were used selectively to build trust in the quantities or levels of testing conducted in safety-critical and mission-critical applications.

PROPOSED MODEL

In this section, a statistical quartile deviation-based reliability estimation model is proposed to find the essential fault factors in the S shaped curves. In this work, a novel approach to predict the software reliability and quality through mean value function using optimal skewness and peakness of the curve. The main objective of this model is to improve the prediction accuracy and to minimize the error rate for software quality and reliability estimation.

Figure 1: Proposed Approach Overview



Optimal Software Reliability Growth Function

*Step*1 : let $\hat{a}$ and $\hat{b}$ are the estimator values

which are real number. Thus $\hat{a}$ and $\hat{b}$ *are*

estimated by using the least square method as

$y = \hat{a}e^{x} + \hat{b}$ --(1)

equation (1) is fitted by using the reliability values.

*step*2 : Let f(x) is the probability density function of continous random varable 'x'.

Q(r) represents the exponential quartile deviation funtion ə $0 \le r \le 1$.

$$\phi_e = \sum \frac{1}{2\pi} e^{-\left(\frac{x-\hat{a}}{\hat{b}}\right)(\mu-\sigma)}$$

μ, σ are the mean, and the standard deviation of variable 'x'

Step 3: Proposed Exponential Quartile density function of current iterative is given by

$$q_e(r) = \phi_e . r^a (1-r)^b \text{ ------(1)}$$

$$q_e(r) = \sum \frac{1}{2\pi} e^{-\left(\frac{x-\hat{a}}{\hat{b}}\right)(\mu-\sigma)} . r^a (1-r)^b$$

$$q_e(r) = m_e r^a (1-r)^b \text{ ------->2}$$

where m_e is constant factor.

Step 4: Exponential Quartile density function of all data is given by

$$Q_e(r) = \int_0^r q_e(r)\,dr \text{------(3)}$$

Step 5:*Exponential Quartile distribution* to all the samples is given by

$$Q(r) = \int_0^r \phi.r^a (1-r)^b\,dr$$

$$Q(r) = \phi.\beta(r, a+1, b+1)$$

$\sin ce\ \beta$ *is* beta *distribution.*

Here r is $r \in (0,1)$, a and b are real values by differentiating equation (1) and substituting q(r)=0

Here we get $u = \dfrac{a}{a+b}$

Equation (1) & (2) should satisfy following constraints, to test the exponential peak and shape of the reliability funtion.

Step 6: The median of the exponential shape curve is given by

$$M\left(Q(r)\right) = Q\left(\frac{1}{2}\right) = \phi_{\frac{1}{2}}\beta\left(\frac{1}{2}, a+1, b+1\right)$$

$$M\left(Q(r)\right) = Q\left(\frac{3}{4}\right) = \phi_{\frac{3}{4}}\beta\left(\frac{3}{4}, a+1, b+1\right)$$

Step 7: Exponential Quartile deviation is given by,

$$EQR = Q_e\left(\frac{3}{4}\right) - Q_e\left(\frac{1}{4}\right)$$

$$= \phi_{\frac{3}{4}}\beta\left(\frac{3}{4}, a+1, b+1\right) - \phi_{\frac{1}{4}}\beta\left(\frac{1}{4}, a+1, b+1\right)$$

Step 8: Exponential coefficeint of skewness of relliability funtionis given by,

$$ES_k = \frac{Q_e\left(\frac{3}{4}\right) + Q_e\left(\frac{1}{4}\right) - 2\left[Q_e\left(\frac{1}{2}\right)\right]}{EQR}$$

$$ES_k = \frac{\phi_{\frac{3}{4}}\beta\left(\frac{3}{4}, a+1, b+1\right) + \phi_{\frac{1}{4}}\beta\left(\frac{1}{4}, a+1, b+1\right) - [Q_e(r, a+1, b+1)]}{Q_e\beta\left(\frac{3}{4}, a+1, b+1\right) - Q_e\beta\left(\frac{1}{4}, a+1, b+1\right)}$$

Step 9: Exponential peaknessof the relliability funtion is given as,

$$EP = \frac{Q_e\left(\frac{7}{8}\right) - Q_e\left(\frac{5}{8}\right) + Q\left(\frac{3}{8}\right) - Q\left(\frac{1}{8}\right)}{EQR}$$

$$EP_k = \beta\left(\frac{7}{8}, a+1, b+1\right) - \beta\left(\frac{5}{8}, a+1, b+1\right) + \beta\left(\frac{3}{8}, a+1, b+1\right) - \beta\left(\frac{1}{8}, a+1, b+1\right)$$

The S-shaped models show the asymptotic behavior similar to the concave model. The failure data used to track the curve are analyzed in two software testing phases. Therefore, the S-shape curve acts in the same way as the concave curve at later testing stages. The test and fault removal process has different approaches and models to determine software reliability. Some models assume that the removal of defects is an exponential distribution according to the time. The goal of adjusting the number of faults is to determine the distribution parameters from which the final SW fault rate can be derived or the estimated. This measure is probabilistic and does not relate to the frequency of failure, which is representative of different modes of failure like systemic SW failures. Initially, different software reliability failure datasets are taken as input to the proposed framework. For each dataset, reliability function is applied to find the mean failure rate on the given input features and parameters as shown in figure 1.

3. Experimental Results

Experimental results are carried out on the software failure datasets taken form the DS1 reported by K.Okumoto. During 56 weeks of testing, a total of 124 faults are identified to test the stability. The second , third and fourth datasets DS2,DS3,DS4 are taken from Rome air development center(RADC) projects.

Figure 2: Mean time to failure rate of the proposed model to the Bernoulli model.

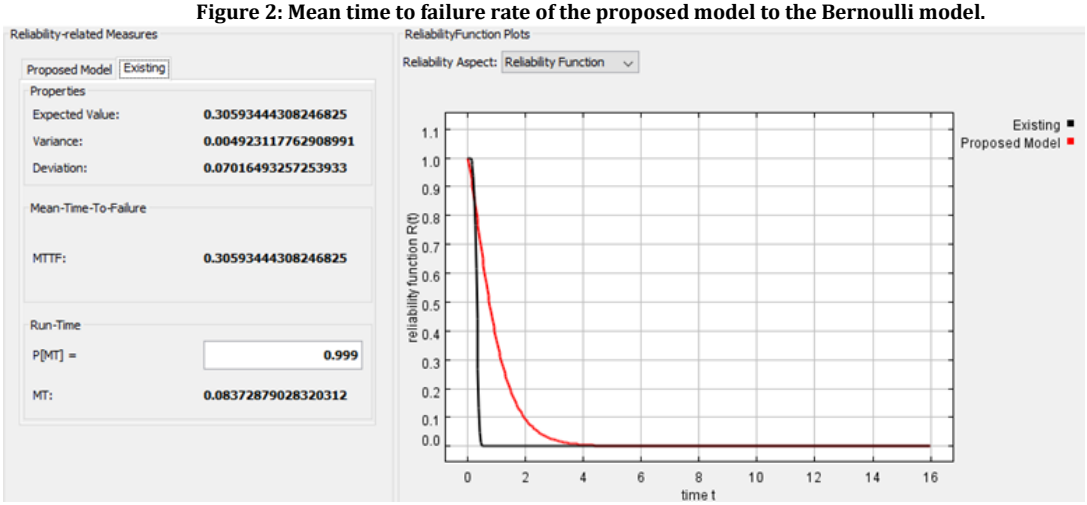


Figure 3: Mean time to failure rate and runtime of the proposed model to the exponential model.

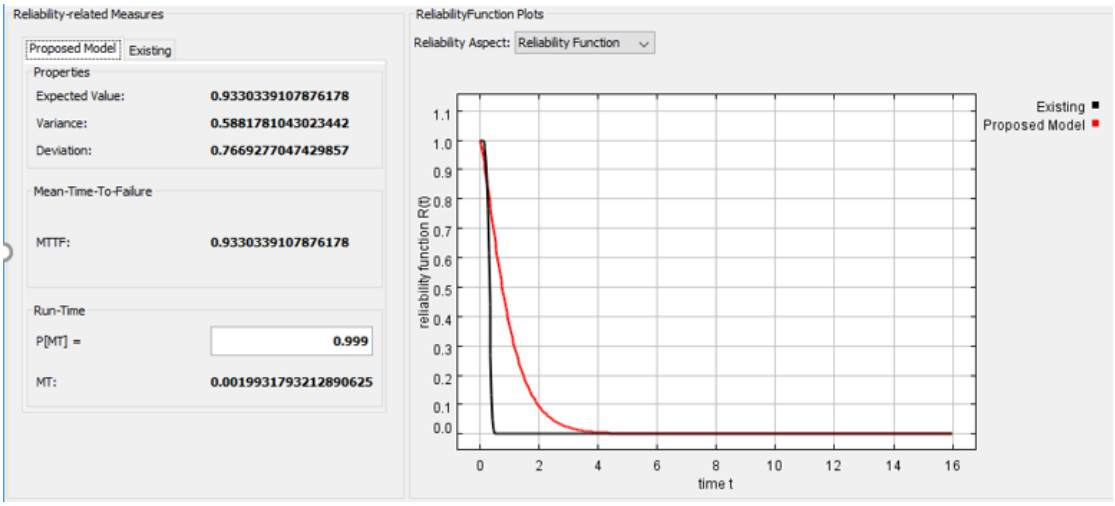


Figure 2, describes the mean time to failure rate of the proposed model to the traditional Bernoulli model on testing data. From the figure, It is clear that the present model has low error rate and better mean time to failure rate than the traditional model.

Figure 3, describes the mean time to failure rate of the proposed model to the traditional exponential model on testing data. From the figure, it is clear that the present model has low error rate and better mean time to failure rate than the traditional model.

Table 1: Performance analysis of MSE rate of the proposed quartile deviation-based reliability estimation model to the traditional models on different reliability datasets.

		MSE			
Datasets	**Rayleigh**	**Exponential**	**Logistic**	**Weibull**	**Proposed Quartile Exponential Function**
DS1	0.45	0.43	0.483	0.49	0.394
DS2	0.425	0.415	0.4266	0.425	0.402
DS3	0.374	0.36	0.396	0.43	0.379
DS4	0.339	0.375	0.361	0.395	0.309

Table 1, describes the performance analysis of proposed quartile exponential software reliability growth model to the traditional models for MSE rate.

CONCLUSION

Software reliability estimation plays a vital role in most of the reliability datasets. Most of the traditional reliability models failed to predict the variation of the faults using the statistical models. The main problem in the existing SRGMs include, difficult to handle large reliability data and these models are not applicable to statistical dependencies and independency measures. In the proposed model, a novel statistical dependency and independency-based quartile density distribution model is implemented to improve the reliability prediction rate. Experimental results proved that the present model has high reliability estimation probability than the traditional growth models in terms of skewness and peak-ness. In the proposed model, the mean rate of prediction ,error rate and runtime are better than the traditional models with different evaluation parameters.

REFERENCES

1. P. K. Kapur, H, Pham, S. Anand , K. Yadav K. "A Unified Approach for Developing Software Reliability Growth Models in the Presence of Imperfect Debugging and Error Generation," IEEE Trans on Reliability, Vol.60, no. 1,331-340,2011
2. P. K. Kapur, A. Aggarwal. G. Kaur .G, "Measuring Concurrent Effect of Time and Testing Coverage using Software Reliability Growth Model with Change Point,"Proceedings of the 5th National Conference; INDIACom,pp613-619, 2011
3. S. Inoue ,S. Yamada , "Two-Dimensional Software Reliability Assessment with Testing-Coverage," Second International Conference on Secure System and Reliability Improvement.,150-157,2008.
4. S. Inoue ,S. Yamada , "Testing-Coverage Dependent Software Reliability Growth Modeling," International Journal of Reliability, Quality and Safety Engineering, Vol. 11, No. 4, 303-312,2004.
5. S. Yamada, M. Obha, S. Osaki, "S-shaped software reliability growth modeling for software error detection," IEEE Trans. On Reliability, Vol. R-32 No. 5, pp. 475-484, 1983.
6. Y. K. Malaiya, M. N. Li, J. M. Bieman, "Software Reliability Growth with Testing Coverage,"IEEE Trans .on Reliability, Vol. 51, No. 4, pp. 420-426, 2002
7. M. Zhong, W. Long, "Whale optimization algorithm with nonlinear control parameter," MATEC Web Conf. vol. 139, pp. 1-5, 2017.
8. C. Zheng, X. Liu, S. Huang, "Estimating parameters of software reliability models by ant colony algorithm," Journal of Computer Applications, vol.32, no.4, pp.1147-1151, 2012.
9. X. Xiao, T. Dohi, "Estimating software reliability using extreme value distribution," Software Engineering, Business Continuity, and Education. Springer Berlin Heidelberg, pp. 399-406, 2011.
10. A. Geol, K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," IEEE Trans. Reliability, vol. 28, pp. 206–211, 1979.
11. L. H. Crow, "Reliability for complex repairable systems," Reliability and Biometry, SIAM, pp. 379–410, 1974.
12. S. Wang, Y. Wu, M. Lu and H. Li, "Discrete Nonhomogeneous Poisson Process Software Reliability Growth Models Based on Test Coverage", .Qual. Reliab. Engng. Int. 2012.
13. J. Iqbal, "Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations","Cogent Engineering (2017), 4: 1286739.
14. K. Rekab, H. Thompson and W. Wu, "An efficient test allocation for software reliability estimation", Applied Mathematics and Computation 220 (2013) 94–103 .
15. R. Garg, K. Sharma, R. Kumar and R. K. Garg, "Performance Analysis of Software Reliability Models using Matrix Method",World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:4, No:11, 2010
16. J. Pati and K. K. Shukla, "A Hybrid Technique for Software Reliability Prediction", "ISEC '15, February 18 - 20, 2015, Bangalore, India" pp. 139-146.