Testing Domain Dependent SRGMS with Power Logistic Function: Analysis and Comparison Using Weighted Criteria Ranking

Dr. Manohar Singh
Associate Professor, PG Dept. of Computer Sc. and IT
Lyallpur Khalsa College, Jalandhar, Punjab, India

ABSTRACT-In this software engineering reliability of software system is considered as the key characteristics of quality of software. Software Reliability Growth Models are helping the software society in predicting and analyzing the software in terms of quality.A Software Reliability Growth Model (SRGM) is one of the basic techniques used to evaluate the software reliability quantitatively.Testing Skill of thesoftwaredesigner affects the testing-domain growth rate and ultimately the number of detectable faults in the system. In this paper, an attempt has been made to describe SRGMs based on Testing Domain with Power Logistic function with and without Testing Skill. This paper also includes the numerical implementation of models using a real time software data set and critical analysis of prediction of Goodness of Fit of Models. In this study, the ranking methodology based on weighted criteria values is used for ranking of SRGMs for comparison.

Keyword- SRGM, Ranking, Comparison Criteria,Goodness-of-Fit, MSE, SSE

I. INTRODUCTION

In this era of technology, developing quality software is essential to compete in the business market. The Traditional model of software quality factors, suggested by McCall, consists of eleven different factors that should be considered in determining the quality of software. Subsequent models include Evans and Marciniak, which consist of twelve factors and Deutsch and Willis, which consists of fifteen factors. All of these models incorporate reliability as one of the software quality factors [1][2][3].A software system is subject to software failures caused by the errors remaining.Reliability of software systems is a key characteristic of software quality.Software reliability is the probability of failure free operationof software in a given period of time under specifiedconditions. Testing is very important in assuring the quality ofthe software by identifying faults in software, and possiblyremoving them[4]. During the testing phase of software development, the software developer executes many test-cases in order to verify functions. These test-cases influence set functions, which is called a testing-domain. The isolated testing-domain expands with increasing number of test-cases and as a result number of detectable faults is also increased. The rate of growth for testing-domain is directly associated to the quality and quantity of the executed test-cases. Testing-skill of the software developer affects the testing-domain growth rate and ultimately the number of detectable faults in the system.Testing-skill of the designer affects the testing-domain growth rate and ultimately the number of detectable faults in the system. In general, there are two cases of test-skills, that is, Low Skill and High Skill. Personnel being inexperienced have lower degree of knowledge of internal structure of the software is meant as Low Skill. Highly experience personnel have high degree of knowledge of internal structure of the software recognized as High Skill. If the test-case designers have low skill, then the number of modules covered by the test-cases is limited and the testing-domain will not spread. On the other hand, if the test-case designers have high level of skill, many modules will be covered by test-cases and the testing-domain will grow to the complete software system. Yamada et al [5][6][7] has done the work on Testing Domain Dependent Software Reliability Models.

This paper is focused on the Critical analysis and the comparison of testing-domain dependent Software Reliability Growth Models with Power Logistic function using weighted criterion ranking method. Organization of this paper consists of five sections. Section II describes the testing-domain dependent SRGMs with Power-Logistic function. Section III includes about Comparison Criteria and weighted criteria ranking methodology of SRGMs. In section IV the parameters of SRGMs are estimated using Brooks &Motley [8]. The goodness-of-fit comparison criteria and ranking of SRGMs using weighted criteria values are also including in section IV. Finding and conclusion are given in section V.

II. DESCRIPTION OF TESTING-DOMAIN DEPENDENT SRGMS

In this section, SRGMs based on the time-dependent behaviour of the testing-domain in the software system is isolates by the executed test cases in software testing using power function and logistic functions are described.

A.Assumptions and Notations for Modeling:

Flexible SRGMs based on the testing-domain which describe software error detection phenomenal during software testing have the following fundamental assumptions:

- A software failure is caused by an error.
- The error causing the failure can be immediately removed.
- Correcting detected errors does not introduce any new error.
- Faults detected exist in the isolated testing-domain.
- No. of faults detected within a small time interval $(t, t + \Delta t)$ is directly proportional to the count of faults left in the testing-domain at testing time t.

The notations used for modeling are:

$m(t)$: expected number of faults identified in the time interval (o, t].

β, k, h: constants.

$b(t)$: fault removal rate as a function of testing time.

$v(t)$: time dependent testing-domain growth rate.

$u_a(t)$, $u_b(t)$: is total number of detectable faults in isolatedtesting domainwithout skill factor at time t.

$u_c(t)$, $u_d(t)$: is total number of faults existing in isolated testing domainwith skill factor at time t.

$m_a(t), m_b(t)$: mean value functions of power logistic models without skillfactor

$m_c(t), m_d(t)$:mean value functions of power logistic models with skill factor

B. Testing domain growth models with Power Logistic function

We will first of all discuss the testing-domain functions for skill factors and without skill factors and finally the reliability growth models based on those testing-domain functions are developed.

a) Flexible Testing-Domain without Skill Factor

From the above assumptions and notations formodeling and taking $v(t)$ to be a power function of testing time, we get the following differential equation:

$$\frac{du_a(t)}{dt} = \lim_{\Delta t \to 0} \frac{u_a(t+\Delta t) - u_a(t)}{\Delta t} = v(t)[a - u_a(t)],$$

$(a > 0, \quad v > 0)$ (1)

Where ais the fault latent in the software.

Under the initial condition$u_a(t = 0) = 0$, we get,

$$u_a(t) = a\left(1 - e^{-v\frac{t^{k+1}}{k+1}}\right) \qquad (2)$$

Where$\left(1 - e^{-v\frac{t^{k+1}}{k+1}}\right)$is the testing-domain growth ratio to the final testing-domain to be covered. For $k = 0, k = 1$ and $k = 2$ the equation takes the shape of Exponential curve, Rayleigh curve and Weibull curve respectively.

Equation assumes that error distribution can be uniform under the condition that $k = 0$ that is, at $v(t) = v$, a constant. In other words, we can say that testing-domain growth rate can be uniform. But, in practice, error distribution is not uniform. As a result, the equation becomes:

$$u_b(t) = a\left(1 - pe^{-v\frac{t^{k+1}}{k+1}}\right), \quad (1 > p > 0) \qquad (3)$$

Where p is the uniformity factor in error distribution.

Let us take power factor k = h - 1 in equations (2) and (3) for simplification, testing domain functions with and without uniformity factor are given as:

$$u_a(t) = a\left(1 - e^{-v\frac{t^h}{h}}\right), \qquad (v > 0) \qquad (4)$$

$$u_b(t) = a\left(1 - pe^{-v\frac{t^h}{h}}\right), \qquad (1 > p > 0) \qquad (5)$$

From assumption that the number of errors detected in the small time interval (t, t - Δt) is proportional to the detectable errors in the isolated testing-domain at a particular time t, we use the differential equation given as:

$$\frac{dm(t)}{dt} = \lim_{\Delta t \to 0} \frac{m(t+\Delta t) - m(t)}{\Delta t}$$

$$\frac{dm(t)}{dt} = b(t)[u(t) - m(t)], (b > 0) \qquad (6)$$

Where $b(t) = \frac{bt^k}{1+\beta e^{-b\frac{t^{k+1}}{k+1}}}$, the power logistic function is used instead of the constant value b, as proposed by Yamada et al. [5] [6] [7], $b(t)$ is the rate of detection of error per remaining error. Power logistic function represents various curve types Exponential, Rayleigh, Weibull. This flexible nature of power logistic function gives the proposed SRGM higher degree of accuracy and wider applicability. Now using testing domain function (2) and (3) and power logistic function b(t) and taking h = k + 1, we get the mean value function of the proposed power logistic SRGMs without skill factors as under:

$$m_a(t) = \frac{a}{1 + \beta e^{\frac{-bt^h}{h}}}\left(1 + \frac{be^{\frac{-vt^h}{h}} - ve^{\frac{-bt^h}{h}}}{v - b}\right), (v > 0)$$

(7)

$$m_b(t) = \frac{a}{1 + \beta e^{\frac{-bt^h}{h}}}\left(1 + \frac{bpe^{\frac{-vt^h}{h}} - (v - b + bp)e^{\frac{-bt^h}{h}}}{v - b}\right)$$

$(1 > p > 0)$ (8)

b) Flexible Testing-Domain with Skill Factor

Testing-skill of the designer affects the testing-domain growth rate and ultimately the number of detectable faults in the system.If the test-case designers have low skill, then the number of modules covered by the test-cases is limited and the testing-domain will not spread. On the other hand, if the test-case designers have high level of skill, many modules will be covered by test-cases and the testing-domain will grow to the complete software system. We have the following differential equations:

$$\frac{ds(t)}{dt} = v[a - s(t)],\ (a > 0, v > 0) \qquad (9)$$

$$\frac{du_c(t)}{dt} = v(t)[s(t) - u_c(t)], \text{where } v(t) = vt^k \qquad (10)$$

Where a is the faults latent in the system, $v(t)$ is the testing-domain growth rate considered to be a function of testing-time t, $s(t)$ is the number of faults detectable at time t and $u_c(t)$ is the number of faults existing in isolated testing-domain at time t. Solving equations simultaneously with initial condition$u_c(t = 0) = 0$, we get:

$$u_c(t) = a\left[1 - \left(1 + v\frac{t^{k+1}}{k+1}\right)e^{-v\frac{t^{k+1}}{k+1}}\right] \qquad (11)$$

Equation (11) shows that testing-domain does not exist at the starting point of testing phase since skill of test-case designer is very low. It also shows that testing-domains growth curve is S-shaped.

Let us now consider the situation where the skill of test-case designer is very high. Persons involved in testing are expert because of which fault detection rate is very high from beginning, and the testing-domain can spread over to the entire software system very quickly. Under the initial condition$u_c(t = 0) = a(1 - p)$, where p is the skill factor of test-case designers, we get:

$$u_d(t) = a\left[1 - p\left(1 + v\frac{t^{k+1}}{k+1}\right)e^{-v\frac{t^{k+1}}{k+1}}\right], (1 > p > 0) \quad (12)$$

Skill factor $p = 0$ indicates that test-case designers are expert and experienced leading to high potential of detecting the faults in initial stages of testing. On the other hand, when $p = 1$, equations signifies that designers have low level of skill. Let us take h = k + 1 in equation (11) and (12) for simplification, testing domain functions with skill factor and with and without uniformity factor are given as:

$$u_c(t) = a\left[1 - \left(1 + \frac{-vt^h}{h}\right)e^{\frac{-vt^h}{h}}\right], (v > 0) \qquad (13)$$

$$u_d(t) = a\left[1 - p\left(1 + \frac{-vt^h}{h}\right)e^{\frac{-vt^h}{h}}\right], (1 > p > 0)(14)$$

As per assumption that the number of errors detected in the small time interval (t, t - Δt) is proportional to the detectable errors in the isolated testing domain at a particular time t, we use the differential equation given as:

$$\frac{dm(t)}{dt} = \lim_{\Delta t \to 0}\frac{m(t+\Delta t) - m(t)}{\Delta t} = b(t)[u(t) - m(t)]$$

Where $b > 0$ *and* $b(t) = \frac{bt^k}{1+\beta e^{-b\frac{t^{k+1}}{k+1}}}$, the power logistic function is used instead of the constant value b, as proposed by Yamada et al. [5] [6] [7], $b(t)$ is the rate of detection of error per error. Power logistic function represents various curve types- Exponential, Rayleigh, Weibull. This flexible nature of power logistic function gives the proposed SRGM higher degree of accuracy and wider applicability.

Now using testing domain function (4.31) and (4.32) and power logistic function b(t) and taking h = k + 1, we get the mean value function of the proposed power logistic SRGMs with skill factors as under:

$$m_c(t) = \frac{a}{1+\beta e^{\frac{-bt^h}{h}}}\left(1 + \frac{b}{v-b}\left(\frac{vt^h}{h} + \frac{2v-b}{v-b}\right)e^{\frac{-vt^h}{h}} - \left(\frac{v}{v-b}\right)^2 e^{\frac{-bt^h}{h}}\right), \ (v \neq b) \quad (15)$$

$$m_d(t) = \frac{a}{1+\beta e^{\frac{-bt^h}{h}}}\left(1 + \frac{bp}{v-b}\left(\frac{vt^h}{h} + \frac{2v-b}{v-b}\right)e^{\frac{-vt^h}{h}} - \left(1 + \frac{bp(2v-b)}{(v-b)^2}\right)e^{\frac{-bt^h}{h}}\right),$$
$$(v \neq b), \quad (1 > p > 0) \qquad (16)$$

The equations (2), (3), (11) and (12) are power testing domain functions suggested by Yamada et al [5][6][7]. These testing domain power functions with skill factor and without skill factor are used to implement by using equations (4), (5), (13) and (14) and named as SRGM-1, SRGM-2, SRGM-3 and SRGM-4. We have proposed four Power Logistic function based Testing-domain Software Reliability Growth Models SRGM-5, SRGM-6, SRGM-7 and SRGM-8, whose mean value functions are given in equations (7), (8), (15) and (16) respectively and are summarized in TABLE 1. We have also compared these models with Yamada et al. [5], basic testing domain model, whose mean value function is given in TABLE 1.

III. COMPARISON CRITERIA AND RANKING METHODOLOGY OF MODELS

A.Comparison Criteria

To investigate the effectiveness and performance of SRGMs, the various predictive and comparison criteria used to compare models quantitatively are R^2 (Coefficient of Multiple Determination), BIAS, VARIANCE, Predictive Ratio Risk (PRR), Accuracy of Estimation (AE), Root Mean Square Predictive Error (RMPSE),Mean Error of Prediction (MEOP),Mean Squared Errors (MSE),Mean Absolute Error (MAE) andSum of Squared Errors (SSE). In all comparison criteria except R^2 is seen that lesser the criterion value provides a better result to the Goodness-of-Fit for SRGM. Model provides a better Goodness-of-Fit for R^2 close to 1[9][10][11]. The summary of comparison criteria is given in TABLE 2.

TABLE 2: Comparison Criteria for Goodness-of-Fit of Models

Comparison Criteria
$MAE = \frac{\sum_{i=1}^{n} \lvert m(t_i) - \hat{m}(t_i)\rvert}{n-p}$ Where p is the number of parameters in model and n represents the time period of testing; $m(t_i), and \hat{m}(t_i)$ are actual and estimated faults corresponding to the time period (t_i)
$MEOP = \frac{\sum_{i=1}^{n} \lvert m(t_i) - \hat{m}(t_i)\rvert}{n-p+1}$
$SSE = \sum_{i=1}^{n} [m(t_i) - \hat{m}(t_i)]^2$
$MSE = \frac{\sum_{i=1}^{n} [m(t_i) - \hat{m}(t_i)]^2}{n}$
$R^2 = 1 - \frac{residualSS}{correctedSS}$
$PRR = \sum_{i=1}^{n} \left[\frac{m(t_i) - \hat{m}(t_i)}{\hat{m}(t_i)}\right]^2$
$BIAS = \frac{\sum_{i=1}^{n} \lvert m(t_i) - \hat{m}(t_i)\rvert}{n}$
$Variation = \sqrt{\frac{\sum_{i=1}^{n} (m(t_i) - \hat{m}(t_i) - BIAS)^2}{n-1}}$
$RMSPE = \sqrt{(Bias^2 + Variation^2)}$
$AE = \left\lvert\frac{M_a - A}{M_a}\right\rvert$, where $M_a and A$ are the actual and estimated cumulative number of detected errors after the test respectively.

B.Weighted Criteria Ranking Methodology

In this section, an attempt is made to develop a quantitative model based on weighted mean tocompute the ranking of Software Reliability Growth Models. In this method of ranking of SRGMs, we use the weight of each comparison criteria; therefore we have named this as the weighted criteria method[13][14]. The steps involved in ranking methodology are:

Step1: Criteria Value Matrix:

Let us consider n numbers of SRGMs having m Comparison Criteria. The Criteria value matrix C is given as:

$$C = \begin{bmatrix} C_{11} & C_{12} & \ldots\ldots & \ldots\ldots & C_{1m} \\ C_{21} & C_{22} & \ldots\ldots & \ldots\ldots & C_{2m} \\ C_{31} & C_{32} & \ldots\ldots & \ldots\ldots & C_{3m} \\ \vdots & \vdots & \ldots\ldots & \ldots\ldots & \vdots \\ C_{n1} & C_{n2} & \ldots\ldots & \ldots\ldots & C_{nm} \\ MINC_1 & MINC_2 & \ldots\ldots & \ldots\ldots & MINC_m \\ MAXC_1 & MAXC_2 & \ldots\ldots & \ldots\ldots & MAXC_m \end{bmatrix}$$

Where, C_{ij} = Value of j^{th} criteria of i^{th} model.

$(MINC)_j$= Minimum value of j^{th} criteria

$(MAXC)_j$= Maximum value of j^{th} criteria, for all i = 1 to n and j = 1 to m.

Step2: Criteria Weighted Matrix:

The Criteria weighted matrix W is given as:

$$W = \begin{bmatrix} W_{11} & W_{12} & \ldots & \ldots & W_{1m} \\ W_{21} & W_{22} & \ldots & \ldots & W_{2m} \\ \vdots & \vdots & \ldots & \ldots & \vdots \\ W_{n1} & W_{n2} & \ldots & \ldots & W_{nm} \end{bmatrix}$$

Where, $W_{ij} = 1 - Z_{ij}$, for i = 1 to n and j = 1 to m.

$Z_{ij}$is criteria rating of j^{th} criteria of i^{th} model. There are two cases to calculate the criteria rating

When the smaller criteria value is best fit to the actual data, the criteria rating is calculated as:

$$Z_{ij} = \frac{(MAXC)_j - C_{ij}}{(MAXC)_j - (MINC)_j}$$

When the larger criteria value is the best fit to the actual data then criteria rating is calculated as:

$$Z_{ij} = \frac{C_{ij} - (MINC)j}{(MAXC)_j - (MINC)_j}$$

Here MAXC meansMaximum value of Criteria and MINC means Minimum value of Criteria.

Step 3: Weighted Criteria Value:

Weighted Criteria value is calculated by multiplying criteria value of each Criterion with their weight. Let $V_{ij}{}^{th}$ is weighted criteria value of j^{th} criteria of i^{th} model and is calculated as:

$$V_{ij} = W_{ij} * C_{ij}$$

The Weighted Criteria value Matrix V is given as:

$$V=\begin{bmatrix} V_{11} & V_{12} & \cdots & \cdots & V_{1m} \\ V_{21} & V_{22} & \cdots & \cdots & V_{2m} \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ V_{n1} & V_{n2} & \cdots & \cdots & V_{nm} \end{bmatrix}$$

Step 4: Permanent Value of Model:

The Weighted mean value of all Predictive Criteria is known as Permanent value of model. The Permanent Value of model is calculated as:

$$P_i = \frac{\sum_{j=1}^{m} V_{ij}}{\sum_{i=1}^{n} W_{ij}}, \text{ for i=1 to n}$$

Step 5: Ranking of Models:

The ranking of models is proposed on the basis of permanent value of the model. The model with smaller permanent value is considered good ranker as compared to the model with bigger permanent value. Thus ranks for all models are provided by comparing permanent values.

IV. NUMERICAL IMPLEMNTATION OF MODELS

In this section we have estimated the parameters of various testing domain models with skill factor and without skill factor using power function and logistic function along with Yamada basic testing domain model using a failure data set. We have also ranked these SRGMs using ranking methodology based on weighted criteria values.

A.Data Description

To check the accuracy of the models as shown in Table 1, a real software fault Data-Set citedfrom Brooks and Motley [8] is used for parameters estimation of models. The fault data set is for a software system of Defense Ground based radar of size 124 KLOC (kilo lines of codes) tested for 35 months in which 1301 faults are identified during 1846.92 hours of testing of Software.

B.Parameter Estimation of Models

The parameters of SRGMs whose mean values functions are given in Table 1 are estimated using Data Set with the help of IBM SPSS Statistical Package. The results of the parameters estimation are shown in TABLE 3.

C.Predictive Criteria of Models

The performance analysis of the models is measured by comparison and predictive criteria of Goodness-of-Fit as in Table 2. The results of the comparison and predictive criteria of goodness of fit for Data-Set are given in Table 3. The graphical representations of goodness of fit curves of proposed and existing models are shown in Figures 1,Figures2, Figures3, Figures4 and Figures5. From these figures, it is clearly indicated that the SRGMs with Power Logistic function fit data set excellently well.

D,Ranking of Models

After evaluation of the goodness of fit comparison and predictive criteria values and performance analysis based on these values, we have also analyzed the models using Ranking Methodology based on weighted criteria values. We have calculated the permanent values of the models after computing weighted criteria value matrix and criteria weighed matrix. Based on permanent values of models, the ranks of the model are calculated. The results are shown in Table 5.

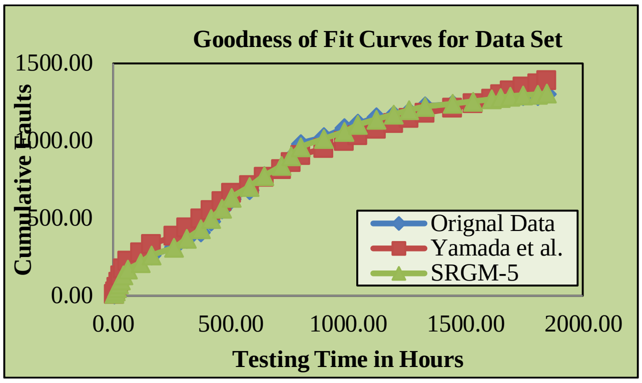

Fig.1: Goodness of Fit Curves of SRGM-5

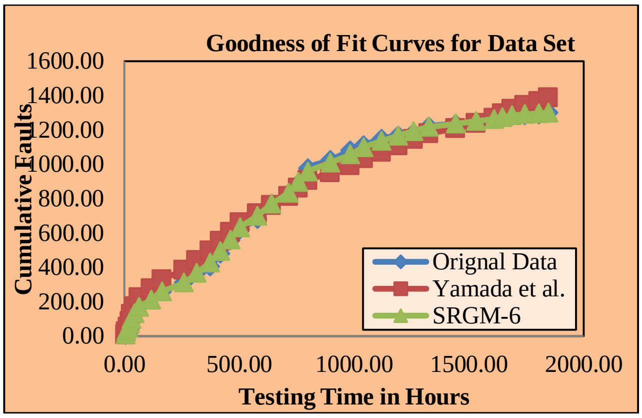

Fig.2: Goodness of Fit Curves of SRGM-6

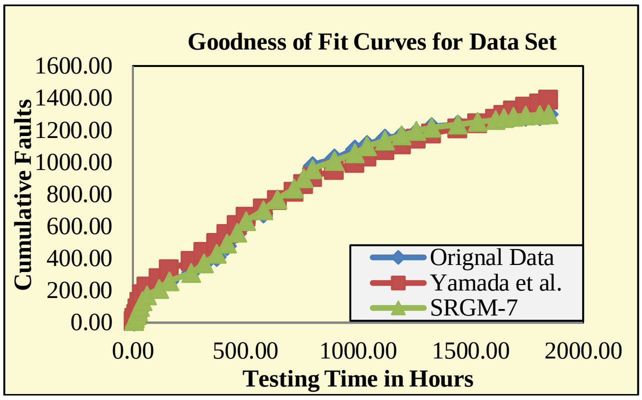

Fig.3: Goodness of Fit Curves of SRGM-7

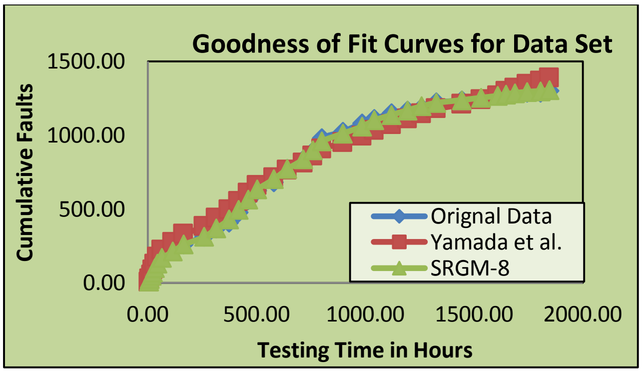

Fig.4: Goodness of Fit Curves of SRGM-8

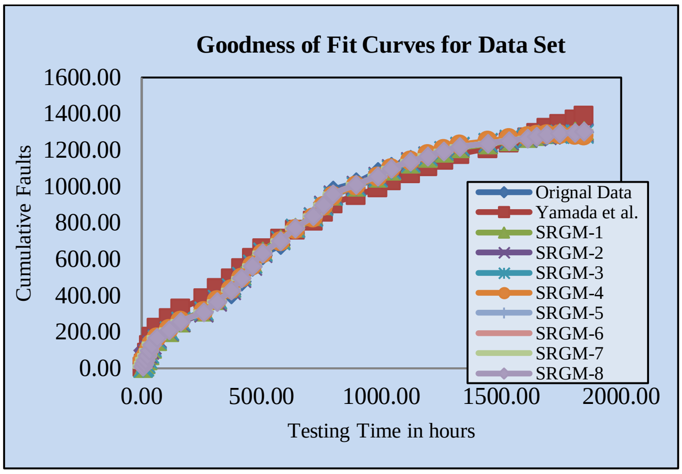

Fig.5: Goodness of Fit Curves of all SRGMs

V. FINDING AND CONCLUSION

It is observed from Table 4 of Comparison and Predictive Criteria MSE, RMPSE, BIAS, AE, MAE and MEOP of the models with Power Logistic Function SRGM-5, SRGM-6, SRGM-7 and SRGM-8 have lower values as compare to Yamada SRGMs and the values of R2of models with Logistic Function are much closed to 1, which clearly indicate the goodness of fit of the proposed models. From Table 5 of permanent values and ranking of models, it is clearly indicated that the SRGMs with Power Logistic function have better ranks as compare to the models SRGM-1, SRGM-2, SRGM-3, SRGM-4 and Yamada et al. model. It is also clearly indicated in graphical illustration of goodness of fit curves of Software Reliability Growth Models of data set shown in Figures 1 to 5 that the SRGM-5, SRGM-6, SRGM-7 and SRGM-8 fit data sets excellently well. It is concluded from the results of the comparison and predictive criteria of SRGMs and permanent values and ranking of SRGMs as shown in Table 4 and Table 5 respectively that the testing domain dependent SRGMswith power logistic function contributes for better performance to estimate software reliability.

VI. REFERENCES

[1]. Alaa F. Sheta and Amal Abdel-Raouf, “Estimating the Parameters of Software Reliability Growth Models using the Grey Wolf Optimization Algorithm”IJACSA, Vol.7, No. 4, 2016.

[2]. M.W. Evans and J. J. Marciniak, Software Quality Assurance and Total Technical Management Approach, New York, USA, John Wiley and Sons, 1987.

[3]. M.S. Deutsch and R. R. Willis, eds., Softwre Quality Engineering, A Total Technical Management Approach, Ch.3. Englewood Cliffs, NJ, USA: Prentice Hall, 1988.

[4]. (DP)P.K. Kapur, Anu G. Aggarwal and Abhishek Tandon, “Two Dimensional Flexible Software Reliability Growth Model with Two Types of Imperfect Debugging”, Proceedings of the 5th National Conference, Computing For Nation Development; INDIACom-2011 ISSN 0973-7529 ISBN 978-93-80544-00-7, March 2011

[5]. Yamada S, Ohtera H and Ohba M, “Testing Domain Dependent Software Reliability Model” Computer Mathematical Applications, 24, pp. 79-86,1992.

[6]. Yamada s and Fujiwara t. “Coverage-Measure and Testing-Domain Metrics based on a software reliability Growth Model", International Journal of Reliability, Quality Safety Engineering. 9(4). 329-340, 2002.

[7]. Yamada S and Fujiwara T. "Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit", International Journal of Reliability, Quality Safety Engineering, 8(3). 205-218, 2001.

[8]. Brooks WD and Motley RW. "Analysis of Discrete Software Reliability Models-Technical Report", (RADC-TR-80-84), New York: Room Air Development Center, 1980.

[9]. Singh M. and Bansal V., ”Parameter Estimation and Validation Testing Procedures for Software Reliability Growth Model”, International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064, 5(12), 1675-1680, 2016.

[10]. Kapur PK, Garg RB and Kumar S., “Contributions to Hardware and Software Reliability”, World Scientific, Singapore, 1999.

[11]. Pillai K and Nair VSs, “A Model for Software Development Efforts and Cost Estimation” IEEE Transactions on Software Engineering, vol. 23 (8), pp. 485-497, 1997.

[12]. Singh M. and Bansal V., “Testing Coverage based Software Reliability Models: Critical Analysis and Ranking based on Weighted Criterion”, International Journal of Computer Science and Information Security (IJCSIS) ISSN No. 1947-5500, 15(2), 2017.

[13]. Mohd. A., Modh. A.H., and Ahmad N., “Analysis and Ranking of Software Reliability Models based on Weighted Criteria value”, I.J. Information Technology and Computer Science”, Vol 2, pp. 1-14, 2013.

[14]. Neha Miglani and Poonam Rana. “Ranking of Software Reliability Growth Models using Greedy Approach,” Global Journal of Business Management and Information Technology. 1 (2). 119-124, 2011.

TABLE 1: Summary of Testing Domain Dependant SRGMsto Evaluate

Model		Mean value function $m(t)$ using Testing Domain
Yamada et al. [5]	Testing Domain Model	$a\left[1-\frac{ve^{-bt}-be^{-vt}}{v-b}\right]$
SRGM-1	Flexible Testing Domain without Skill Factor	$a\left(1-e^{-v\frac{t^h}{h}}\right)$
SRGM-2	Flexible Testing Domain without Skill Factor with uniformity Factor	$a\left(1-pe^{-v\frac{t^h}{h}}\right)$
SRGM-3	Flexible Testing Domain with Skill Factor	$a\left[1-\left(1+\frac{-vt^h}{h}\right)e^{\frac{-vt^h}{h}}\right]$
SRGM-4	Flexible Testing Domain with Skill Factor with uniformity Factor	$a\left[1-p\left(1+\frac{-vt^h}{h}\right)e^{\frac{-vt^h}{h}}\right]$
SRGM-5	Proposed Power Logistic Testing domain model without skill factor	$\frac{a}{1+\beta e^{\frac{-bt^h}{h}}}\left(1+\frac{be^{\frac{-vt^h}{h}}-ve^{\frac{-bt^h}{h}}}{v-b}\right)$
SRGM-6	Proposed Power Logistic Testing domain model without skill factor	$\frac{a}{1+\beta e^{\frac{-bt^h}{h}}}\left(1+\frac{bpe^{\frac{-vt^h}{h}}-(v-b+bp)e^{\frac{-bt^h}{h}}}{v-b}\right)$
SRGM-7	Proposed Power Logistic Testing domain model with skill factor	$\frac{a}{1+\beta e^{\frac{-bt^h}{h}}}\left(1+\frac{b}{v-b}\left(\frac{vt^h}{h}+\frac{2v-b}{v-b}\right)e^{\frac{-vt^h}{h}}-\left(\frac{v}{v-b}\right)^2e^{\frac{-bt^h}{h}}\right)$
SRGM-8	Proposed Power Logistic Testing domain modelwith skill factor	$\frac{a}{1+\beta e^{\frac{-bt^h}{h}}}\left(1+\frac{bp}{v-b}\left(\frac{vt^h}{h}+\frac{2v-b}{v-b}\right)e^{\frac{-vt^h}{h}}-\left(1+\frac{bp(2v-b)}{(v-b)^2}\right)e^{\frac{-bt^h}{h}}\right)$

TABLE 3: Parameters Estimation of the SRGMs

Model	**Estimated Parameters**					
	a	**v**	**h**	**b**	**β**	**p**
Yamada et al.	1689.369	.090		.090		
SRGM-1	1325.3671	0.0036	2.2266			
SRGM-2	1303.8585	0.0018	2.5012			0.9588
SRGM-3	1137.3953	0.0026	2.1001			
SRGM-4	1137.4592	0.0017	2.2670			0.9633
SRGM-5	1321.9453	37.9197	1.1797	0.119	10.6927	
SRGM-6	1321.8169	79.256	1.1812	0.1186	10.6609	0.6430
SRGM-7	1321.9545	47.8124	1.1815	.1184	10.6105	
SRGM-8	1322.186	18.616	1.180	.119	10.605	0.793

TABLE 4: Comparison and Predictive Criteria of the SRGMs

MODEL	**R^2**	**MSE**	**BIAS**	**VARIANCE**	**RMPSE**	**MAE**	**MEOP**	**AE**
Yamada et al.	0.987	2967.582	43.76	255.18	258.91	47.87	46.42	0.067
SRGM-1	0.997	638.932	18.47	107.72	109.3	20.21	19.59	0.006
SRGM-2	0.999	306.859	12.48	72.77	73.83	13.65	13.24	0.003
SRGM-3	0.998	518.063	16.28	94.91	96.3	17.8	17.26	0.009
SRGM-4	0.999	316.916	13.27	77.38	78.5	14.51	14.07	0.018
SRGM-5	0.999	223.874	10.19	59.43	60.3	11.89	11.51	0.002
SRGM-6	0.999	230.225	10.16	59.25	60.12	12.26	11.86	0.001
SRGM-7	0.999	225.006	10.21	59.55	60.42	11.91	11.53	0.002
SRGM-8	0.999	236.153	10.28	59.95	60.83	12.41	12	0.002

Table 5: The Permanent Values and Ranking of Models

Model	Sum of Weight	Sum of Weighted Values	Permanent Value	Rank
Yamada et al.	8.0000	3620.7760	452.5970	9
SRGM-1	1.5985	164.2861	102.7755	8
SRGM-2	0.3660	21.5800	58.9555	6
SRGM-3	1.1869	99.1652	83.5509	7
SRGM-4	0.7152	28.4865	39.8307	5
SRGM-5	0.0179	0.1183	6.6221	1
SRGM-6	0.0226	0.7779	34.3836	4
SRGM-7	0.0212	0.3036	14.3085	2
SRGM-8	0.0588	1.8728	31.8342	3