Analysis of Some Software Reliability Growth Models Using Failure Data of Real Time Control System

Javaid Iqbal

Department of Computer Sciences, University of Kashmir, Srinagar, 190006, India
Email: iamjavaid@gmail.com

***Abstract*— A newly developed software system is subjected to vigorous testing before its deployment. Testing aims at minimizing the probability of occurrence of failures. The aim of testing process is to build confidence in the software for its use in real world applications. Thus, reliability of systems is always an important issue for us. Growth in reliability takes place as the error detection and correction process is employed on the software. In order to model reliability growth, many formulations called as Software Reliability Growth Models (SRGMs) have been proposed that include some SRGMs based on Non-Homogeneous Poisson Process (NHPP). The role of human learning and experiential pattern gains are being studied and incorporated in such models. The realistic assumptions about human learning behavior and experiential gains of new skill-sets for better detection and correction of faults on software are being incorporated and studied in such models. In this paper, an analysis of some select SRGMs with learning effects is presented based on use of failure data of real time control system. Moreover, model comparisons on the basis of total defects predicted by the select models are also presented.

Index Terms—Software Reliability, Software Reliability Growth Model (SRGM), Non-Homogeneous Poisson Process (NHPP), Learning effect, two-type learning effect.

I. INTRODUCTION

Many software reliability growth models (SRGMs) have been proposed in literature. Many have been proposed under the analytical framework of a Non-Homogeneous Poisson Process (NHPP).The aim is to better model the error-detection and correction processes by trying to incorporate some realistic underlying assumptions. Goel and Okumoto in [1] proposed an exponential SRGM. Yamada and Ohba in [2] proposed delayed S-shaped SRGM while Ohba in [3] proposed inflection S-shaped SRGM. Gokhale and Trivedi in [4] proposed an enhanced NHPP model which takes into account the time-dependent failures occurring in debugging process. Debugging process can be perfect or imperfect. Debugging process based on the assumption that that each time an error occurred, the fault that caused it can be removed immediately without introduction of new faults is called perfect.

Presently imperfect debugging is in vogue, which is based on a more realistic assumption that the removal of a fault can introduce some new faults [5, 6]. An insight into imperfect debugging can be found in Obha [3,7], Pham[8], Kapur and Younes [9], Shyur [10] and Chiu and Huang [11]. Other realistic assumptions for running environment, testing/debugging strategies and resource allocation can also affect the reliability growth as discussed in Chiu and Huang [11] and Shyur [10]. Many researchers have used NHPP based SRGMs to capture the reliability growth of a software from the processes of testing and debugging [23-28]. Recently, a unified framework for use of SRGMs with learning process and error generation in imperfect debugging environments has been presented in [12]. Chiu and Huang in [11] proposed a learning-effect-based NHPP model that captures the learning effect gained by testing/ debugging staff from inspection and debugging of code. In [13,14] Chiu proposes an improvement model under time-dependent learning effect. In [5] Iqbal, Ahmad and Quadri propose an SRGM that incorporates two types of learning effects and then incorporate a negligence factor also into the SRGM with two types of learning effect in [6]. They basically indicate that the two types of learning effect are autonomous learning and acquired learning with acquired learning gained after a spell of repeated experience/observation of the testing/debugging process by the tester/debugger resulting in concept formation by the tester/debugger about that particular pattern. Recently in [29, 30,31] learning based fault detection rates have been incorporated in imperfect debugging models. In this paper, we refine the definition of autonomous learning as the assimilation of know-how by doing (testing) without role of experience and the acquired learning is refined to the definition of learning that stands acquired after a spell of repeated experience/observation of the testing/debugging process by the tester/debugger.

The rest of the paper is organized as: Section II introduces the non-homogeneous Poisson process. Section III discusses how some select learning based models evolved by improvements starting from the learning model proposed by Chiu and Huang in [11], through improvements by Chiu[13] , Iqbal et al [5] introduced the concept of two types of learning in SRGM and later improved it in [6]. This progression in SRGM development is discussed in section III. Section IV presents parameter estimation and presents a comparison of models for total defects predicted using failure data from real time control system.

II. NHPP MODELING CONCEPTS

As an error counting process $\{N(t), t\geq 0\}$ with mean $m(t)$ and failure intensity rate $\lambda(t)$ a general NHPP process is written mathematically as:

$$Pr(N(t)=k) = \frac{[m(t)^k e^{-m(t)}]}{k!},\ k = 0,1,2,3,....$$

with mean value function m(t) representing the expected number of errors detected within time (0,t) and mathematically represented as an integral of intensity function between time zero(start) and time t. The conditional software reliability R(s/t) which is the probability that no error is detected within a specific time interval (t, t+s), given that an error has occurred at time t (t≥ 0, s>0) and is mathematically written as

$$R(s/t) = e^{-[m(t+s)-m(t)]}$$

with limiting value of $R(s/t) \approx 1$ as time approaches to infinity.

III. DEVELOPMENTAL PROGRESSION OF SOME LEARNING-BASED MODELS

Here we present a brief account of the progression of development of some select learning based SRGMs.

A. Chiu and Huang Learning Model [11]:

A learning factor η that arises from inspection of the testing/debugging codes under the assumption that η does not change with time is considered.

Model equation is

$$f(t) = \frac{dF(t)}{dt} = \big(\alpha + \eta F(t)\big)\big(1 - F(t)\big)$$

where autonomous error factor $\alpha > 0$ and learning factor $\eta > 0$.

The explicit solution of $F(t)$ is given by:

$$F(t) = 1 - \frac{1 + (\eta/\alpha)}{(\eta/\alpha) + e^{(\alpha+\eta)t}}$$

and

$$f(t) = \frac{(\alpha+\eta)^2 e^{(\alpha+\eta)t}}{\alpha\left((\eta/\alpha) + e^{(\alpha+\eta)t}\right)^2}$$

where mean value function $m(t)$ is $m(t) = aF(t)$

$$m(t) = a\left\{1 - \frac{1+(\eta/\alpha)}{(\eta/\alpha)+e^{(\alpha+\eta)t}}\right\} \qquad (1)$$

intensity function $\lambda(t) = \frac{d(m(t))}{dt} = af(t)$

and error detection rate is

$$d(t) = \frac{\lambda(t)}{a-m(t)} = (\alpha+\eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta)t}+\eta}\right)$$

B. Chiu Improvement Model [13,14]:

A learning factor η that arises from inspection of the testing/debugging codes under the assumption that η does not change with time and a negligent factor τ, that arises from negligence on part of testers/developers in correcting errors from learnt patterns previously detected, are considered. Model equation is

$$f(t) = \frac{dF(t)}{dt} = \big(\alpha + \eta F(t) - \tau\big)\big(1 - F(t)\big)$$

The explicit solution of $\boldsymbol{F(t)}$is given by:

$$\boldsymbol{F(t) = 1 - \frac{1 + \left(\eta/_{(\alpha-\tau)}\right)}{\left(\eta/_{(\alpha-\tau)}\right) + e^{(\alpha+\eta-\tau)t}}}$$

and

$$\boldsymbol{f(t) = \frac{(\alpha+\eta)^2 e^{(\alpha+\eta)t}}{(\alpha-\tau)\left(\left(\eta/_{(\alpha-\tau)}\right) + e^{(\alpha+\eta-\tau)t}\right)^2}}$$

Where mean value function$\boldsymbol{m(t)}$is$\boldsymbol{m(t) = aF(t)}$

$$\boldsymbol{m(t) = a\left\{1 - \frac{1+\left(\eta/_{(\alpha-\tau)}\right)}{\left(\eta/_{(\alpha-\tau)}\right) + e^{(\alpha+\eta-\tau)t}}\right\}} \qquad (2)$$

, intensity function $\boldsymbol{\lambda(t) = \frac{d(m(t))}{dt} = af(t)}$ and error detection rate is

$$\boldsymbol{d(t) = \frac{\lambda(t)}{a-m(t)} = (\alpha - \tau + \eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta-\tau)t} + \eta}\right)}$$

C. A Two-Type Learning Model [5]:

Two type of learning effect, which are autonomous learning η_1 and acquired learning η_2 which represents experiential gains in learning are considered.

Model equation is

$$\boldsymbol{f(t) = \frac{dF(t)}{dt} = \left(\eta_1\alpha + \eta_2 F(t)\right)\left(1 - F(t)\right)}$$

where autonomous error factor, $\boldsymbol{\alpha > 0}$, type-I learning factor(autonomous learning) $\boldsymbol{\eta_1}$> 0 and type-II learning factor(acquired learning) $\boldsymbol{\eta_2}$> 0. The explicit solution of $\boldsymbol{F(t)}$ is given by

$$\boldsymbol{F(t) = 1 - \frac{1 + \left(\eta_2/_{\eta_1\alpha}\right)}{\left(\eta_2/_{\eta_1\alpha}\right) + e^{(\eta_1\alpha+\eta_2)t}}}$$

and

$$\boldsymbol{f(t) = \frac{(\eta_1\alpha+\eta_2)^2 e^{(\eta_1\alpha+\eta_2)t}}{\alpha\left(\left(\eta_2/_{\eta_1\alpha}\right) + e^{(\eta_1\alpha+\eta_2)t}\right)^2}}$$

The mean value function $\boldsymbol{m(t)}$*is*

$$\boldsymbol{m(t) = aF(t) = a\left\{1 - \frac{1+\left(\eta_2/_{\eta_1\alpha}\right)}{\left(\eta_2/_{\eta_1\alpha}\right) + e^{(\eta_1\alpha+\eta_2)t}}\right\}}, \qquad (3)$$

the intensity function is

$$\boldsymbol{\lambda(t) = \frac{d(m(t))}{dt} = af(t) = a\left\{\frac{(\eta_1\alpha+\eta_2)^2 e^{(\eta_1\alpha+\eta_2)t}}{\alpha\left(\left(\eta_2/_{\eta_1\alpha}\right) + e^{(\eta_1\alpha+\eta_2)t}\right)^2}\right\}}$$

and variation in error detection rate per error at time $\mathbf{t}$ is given by

$$\boldsymbol{d(t) = \frac{\lambda(t)}{a - m(t)}}$$

$$= \boldsymbol{(\eta_1\alpha + \eta_2)\left(1 - \frac{\eta_2}{\eta_1\alpha e^{(\eta_1\alpha+\eta_2)t} + \eta_2}\right)}$$

D. ATwo-Type Learning Model with negligence factor[6]

Two types of learning effect, which are autonomous learning η_1 and acquired learning η_2 which represents experiential gains in learning and a negligence factor τ that arises from the negligence on part of testers/developers in correcting errors from learnt patterns previously detected are considered.

The model equation is:

$$\boldsymbol{f(t) = \frac{dF(t)}{dt} = \left(\eta_1\alpha + \eta_2 F(t) - \tau\right)\left(1 - F(t)\right)}$$

where $\boldsymbol{\alpha > 0}$, $\boldsymbol{\eta_1}$*> 0* and$\boldsymbol{\eta_2}$*> 0.*

The explicit solution of $\boldsymbol{F(t)}$is given by

$$F(t) = 1 - \frac{1 + \left(\eta_2/(\eta_1\alpha - \tau)\right)}{\left(\eta_2/(\eta_1\alpha - \tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}}$$

and

$$f(t) = \frac{(\eta_1\alpha + \eta_2 - \tau)^2 e^{(\eta_1\alpha+\eta_2-\tau)t}}{\alpha\left(\left(\eta_2/(\eta_1\alpha - \tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}\right)^2}$$

The mean value function$m(t)$is

$$m(t) = aF(t)$$

$$m(t) = a\left\{1 - \frac{1+\left(\eta_2/(\eta_1\alpha-\tau)\right)}{\left(\eta_2/(\eta_1\alpha-\tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}}\right\} \quad (4)$$

whereas the intensity function

$$\lambda(t) = \frac{d(m(t))}{dt} = af(t) = a\left\{\frac{(\eta_1\alpha + \eta_2 - \tau)^2 e^{(\eta_1\alpha+\eta_2-\tau)t}}{\alpha\left(\left(\eta_2/(\eta_1\alpha - \tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}\right)^2}\right\}$$

and variation in error detection rate per error at time is given by

$$d(t) = \frac{\lambda(t)}{a - m(t)}$$

$$= (\eta_1\alpha + \eta_2 - \tau)\left(1 - \frac{\eta_2}{(\eta_1\alpha-\tau)e^{(\eta_1\alpha+\eta_2-\tau)t} + \eta_2}\right)$$

III. PARAMETER ESTIMATION

Fitting the proposed models to the actual data is done by estimating the model parameters. We have used SPSS to estimate the model parameters by using Regression under Non-linear mode. The mean value functions represented in equations (1) to (4) are used in estimation of parameters. The parameter estimation is done using failure data from Real Time Control System dataset. Real Time Control System Dataset has recorded a total fault count of 136 against the time-between-failure (TBF) measured in seconds[20]

The following table-2 presents the mean value functions and FDRs of some select models.

Table-2 : Model Names and Mean value Function

Goel Okumotto [1]

$m(t) = a(1- e^{-bt})$

Chiu and Huang Learning Model [11]

$$m(t) = a\left\{1 - \frac{1 + (\eta/\alpha)}{(\eta/\alpha) + e^{(\alpha+\eta)t}}\right\}$$

2-type Learning Model-1(2TL1) [5]

$$m(t) = a\left\{1 - \frac{1 + (\eta_2/\eta_1\alpha)}{(\eta_2/\eta_1\alpha) + e^{(\eta_1\alpha+\eta_2)t}}\right\}$$

2-type Learning Model-2 (2TL2) [6]

$$m(t) = a\left\{1 - \frac{1 + \left(\eta_2/(\eta_1\alpha - \tau)\right)}{\left(\eta_2/(\eta_1\alpha - \tau)\right) + e^{(\eta_1\alpha+\eta_2-\tau)t}}\right\}$$

The following table presents the values of parameters of select models using failure data from Real Time Control System dataset using mean value functions listed in table 2.

Table-3: Estimation of parameters under failure data from Real Time Control System dataset for select models

Comparison under dataset[6]	
Model	Parameters
G-O[1]	a= 124.44, b=.051
Chiu[11]	a=124.171, α =.051, η=.001
2TL1[5]	a=124.171, α =.001, η1 =88.486, η2 =.001
2TL2[6]	a=124.437, α =.163, η1 =5.874, η2 =1.000E-5, τ =.909

V. Results and Comparative Analysis

There are many comparison criteria as defined in [22] wherein the authors have presented analysis and ranking of software reliability models based on weighted criteria. The comparison criteria used is R^2 measure also called as coefficient of multiple determinations (R^2) which is usually used to depict the goodness-of-fit and is expressible as: [15]

$$\boldsymbol{R^2 = 1 - \frac{(ResidualSumofSquares)}{(CorrectedSumofSquares)}}$$

$$\boldsymbol{R^2 = 1 - \frac{\sum_{i=1}^{n}(m_i - m(t_i))^2}{\sum_{i=1}^{n}(m_i - \sum_{k=1}^{n} {m_k}/{n})^2}}$$

$\boldsymbol{R^2}$ represents a measure of the percentage of the total variation about the mean for the fitted curve. It lies in the range of 0 to 1, with a larger $\boldsymbol{R^2}$ value indicating a better representation of variation about the mean of the data set by the model equation. However, a smaller $\boldsymbol{R^2}$value indicates that the model equation fails to represent the variations in the data set. Obviously, a near-one value of $\boldsymbol{R^2}$ is highly desirable [15]. A comparative analysis of some select models using $\boldsymbol{R^2}$ measure is presented using seven datasets and model comparisons on the basis of total defects predicted by the select models are also tabulated.

The following table-4 presents the results of goodness-of-fit under R^2comparison criteria

Table-4: Goodness-of-fit under R^2comparison criteria using failure data from Real Time Control System dataset

Comparison under R-sq for failure data from Real Time Control System dataset				
Dataset	G-O [1]	Chiu[11]	2TL1[5]	2TL2 [6]
[6]	.978	.977	.977	.978

The following table presents the comparison of models for total defects predicted under Failure data of real time control system

Table-5: Comparison of models for total defects predicted using failure data from Real Time Control System dataset:

Testing time	**Defects**	**G-O**	**Chiu**	**2TL1**	**2TL 2**
0.003	1	0.018976	0.018894	0.018894	0.018976
0.033	2	0.208582	0.207684	0.207684	0.208572
0.146	3	0.920171	0.916267	0.916267	0.920131
0.227	4	1.427739	1.421744	1.421744	1.427677
0.342	5	2.14478	2.135907	2.135907	2.144689

0.351	6	2.200719	2.191626	2.191626	2.200626
0.353	7	2.213147	2.204004	2.204004	2.213053
0.444	8	2.777265	2.765928	2.765928	2.777149
0.556	9	3.467992	3.454041	3.454041	3.467848
0.571	10	3.560202	3.545909	3.545909	3.560055
0.709	11	4.40524	4.387875	4.387875	4.405062
0.759	12	4.709954	4.69151	4.69151	4.709764
0.836	13	5.1777	5.157632	5.157632	5.177494
0.86	14	5.323118	5.302551	5.302551	5.322906
0.968	15	5.975304	5.95255	5.95255	5.97507
1.056	16	6.504074	6.479599	6.479599	6.503822
1.726	17	10.4533	10.41744	10.41744	10.45293
1.846	18	11.14653	11.10893	11.10893	11.14614
1.872	19	11.29618	11.25821	11.25821	11.29578
1.986	20	11.94997	11.91046	11.91046	11.94957
2.311	21	13.79321	13.74969	13.74969	13.79276
2.366	22	14.10214	14.05801	14.05801	14.10169
2.608	23	15.45122	15.40456	15.40456	15.45074
2.676	24	15.82732	15.78001	15.78001	15.82683
3.098	25	18.13253	18.08168	18.08168	18.132
3.278	26	19.10084	19.04876	19.04876	19.1003
3.288	27	19.15438	19.10223	19.10223	19.15384
4.434	28	25.11279	25.05609	25.05609	25.1122
5.034	29	28.09666	28.03957	28.03957	28.09606
5.049	30	28.17009	28.11301	28.11301	28.1695
5.085	31	28.34611	28.28904	28.28904	28.34552
5.089	32	28.36565	28.30858	28.30858	28.36505
5.089	33	28.36565	28.30858	28.30858	28.36505
5.097	34	28.40472	28.34765	28.34765	28.40412
5.324	35	29.50656	29.4497	29.4497	29.50596
5.389	36	29.81973	29.76296	29.76296	29.81913
5.565	37	30.66252	30.60605	30.60605	30.66193
5.623	38	30.93861	30.88226	30.88226	30.93802
6.08	39	33.08578	33.03068	33.03068	33.0852
6.38	40	34.46841	34.41443	34.41443	34.46784
6.477	41	34.91097	34.8574	34.8574	34.91041
6.74	42	36.09998	36.04761	36.04761	36.09943
7.192	43	38.10668	38.05672	38.05672	38.10615
7.447	44	39.2186	39.17017	39.17017	39.21808
7.644	45	40.06779	40.02063	40.02063	40.06729
7.837	46	40.89154	40.84568	40.84568	40.89105
7.843	47	40.91702	40.8712	40.8712	40.91653
7.922	48	41.25177	41.20651	41.20651	41.25129
8.738	49	44.63196	44.5929	44.5929	44.63154
10.089	50	49.92908	49.90193	49.90193	49.92877
10.237	51	50.48757	50.46181	50.46181	50.48727

10.258	52	50.56647	50.54091	50.54091	50.56618
10.491	53	51.43632	51.41299	51.41299	51.43605
10.625	54	51.93192	51.90989	51.90989	51.93166
10.982	55	53.23595	53.21741	53.21741	53.23572
11.175	56	53.93113	53.9145	53.9145	53.93092
11.411	57	54.77198	54.75771	54.75771	54.7718
11.442	58	54.88168	54.86772	54.86772	54.8815
11.811	59	56.17431	56.16407	56.16407	56.17417
12.559	60	58.72136	58.71874	58.71874	58.7213
12.559	61	58.72136	58.71874	58.71874	58.7213
12.791	62	59.49188	59.49163	59.49163	59.49184
13.121	63	60.57234	60.57546	60.57546	60.57233
13.486	64	61.74647	61.7533	61.7533	61.7465
14.708	65	65.52253	65.54162	65.54162	65.52268
15.251	66	67.12662	67.15101	67.15101	67.12683
15.261	67	67.15575	67.18023	67.18023	67.15595
15.277	68	67.20232	67.22696	67.22696	67.20253
15.806	69	68.72104	68.75072	68.75072	68.72129
16.185	70	69.78427	69.81748	69.81748	69.78457
16.229	71	69.90639	69.94	69.94	69.90668
16.358	72	70.26283	70.29763	70.29763	70.26314
17.168	73	72.44836	72.49038	72.49038	72.44874
17.458	74	73.20921	73.25372	73.25372	73.20961
17.758	75	73.98458	74.03161	74.03161	73.98501
18.287	76	75.32333	75.37468	75.37468	75.32381
18.568	77	76.01996	76.07352	76.07352	76.02046
18.728	78	76.41219	76.46699	76.46699	76.4127
19.556	79	78.3918	78.45272	78.45272	78.39237
20.567	80	80.69863	80.76634	80.76634	80.69927
21.012	81	81.67702	81.74749	81.74749	81.67769
21.308	82	82.31567	82.38788	82.38788	82.31636
23.063	83	85.91104	85.99225	85.99225	85.91183
24.127	84	87.93966	88.02516	88.02516	87.94049
25.91	85	91.1025	91.19331	91.19331	91.10339
26.77	86	92.52855	92.62108	92.62108	92.52946
27.753	87	94.084	94.17786	94.17786	94.08492
28.46	88	95.15563	95.25004	95.25004	95.15656
28.493	89	95.20472	95.29915	95.29915	95.20564
29.361	90	96.46667	96.56131	96.56131	96.4676
30.085	91	97.47749	97.57194	97.57194	97.47842
32.408	92	100.4807	100.5724	100.5724	100.4816
35.338	93	103.7962	103.8806	103.8806	103.7971
36.799	94	105.2739	105.3531	105.3531	105.2747
37.642	95	106.0779	106.1538	106.1538	106.0786
37.654	96	106.0891	106.1649	106.1649	106.0898
37.915	97	106.3309	106.4057	106.4057	106.3317

39.715	98	107.9144	107.9811	107.9811	107.9151
40.58	99	108.6253	108.6878	108.6878	108.626
42.015	100	109.7379	109.793	109.793	109.7384
42.045	101	109.7603	109.8152	109.8152	109.7608
42.188	102	109.8666	109.9208	109.9208	109.8672
42.296	103	109.9464	110	110	109.947
42.296	104	109.9464	110	110	109.947
45.406	105	112.0658	112.1018	112.1018	112.0662
46.653	106	112.8259	112.8543	112.8543	112.8262
47.596	107	113.3695	113.3922	113.3922	113.3697
48.296	108	113.7565	113.7749	113.7749	113.7567
49.171	109	114.2212	114.2342	114.2342	114.2214
49.416	110	114.3477	114.3591	114.3591	114.3479
50.145	111	114.7149	114.7217	114.7217	114.715
52.042	112	115.6089	115.6037	115.6037	115.6089
52.489	113	115.8073	115.7992	115.7992	115.8072
52.875	114	115.975	115.9645	115.9645	115.9749
53.321	115	116.1648	116.1514	116.1514	116.1647
53.443	116	116.2159	116.2018	116.2018	116.2158
54.433	117	116.6196	116.5992	116.5992	116.6194
55.381	118	116.9875	116.9611	116.9611	116.9873
56.463	119	117.3863	117.3531	117.3531	117.386
56.485	120	117.3942	117.3609	117.3609	117.3939
56.56	121	117.421	117.3872	117.3872	117.4207
57.042	122	117.5909	117.5541	117.5541	117.5906
62.551	123	119.2637	119.1935	119.1935	119.2631
62.651	124	119.29	119.2191	119.2191	119.2893
62.661	125	119.2926	119.2217	119.2217	119.2919
63.732	126	119.5653	119.4882	119.4882	119.5646
64.103	127	119.6564	119.5772	119.5772	119.6557
64.893	128	119.8447	119.761	119.761	119.8439
71.043	129	121.0783	120.962	120.962	121.0772
74.364	130	121.6005	121.4683	121.4683	121.5992
75.409	131	121.7474	121.6105	121.6105	121.7461
76.057	132	121.8346	121.6948	121.6948	121.8333
81.542	133	122.4685	122.3063	122.3063	122.4669
82.702	134	122.5814	122.4149	122.4149	122.5798
84.566	135	122.7494	122.5763	122.5763	122.7477
88.682	136	123.0685	122.882	122.882	123.0667

Table 5: Comparison of proposed 2-TL models with select models for total defects predicted using failure data from Real Time Control System dataset

IV. CONCLUSION

In this paper, a detailed analysis of some select SRGMs with learning effects is presented, the developmental progression is shown. Failure data from Real Time Control System dataset has been used for this detailed analysis and parameter estimation is also presented. The parameters estimated are competing with other famous models. R^2comparison criteria shows fairly good values to validate the models. Finally, model comparisons on the basis of total defects predicted by the select models are also presented in table using failure data from Real Time Control System dataset.

REFERENCES

[1] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software and other performance measures,"IEEE Transactions on Reliability, vol. 28, pp. 206–211, 1979.

[2] S. Yamada and H. Ohba "S-shaped software reliability modeling for software error detection", 1983, IEEE Trans Reliab;32:475–84.

[3] M. Ohba, "Inflexion S-shaped software reliability growth models", Stochastic Models in Reliability Theory (S. Osaki, Y. Hatoyama, Eds), 1984, pp 144 - 162; Springer- Verlag Merlin.

[4] S.S. Gokhale, and K.S.Trivedi "A time/structure based software reliability model" Ann Software Eng; 1999, 8:85–121.

[5] J. Iqbal, N. Ahmad, and S.M.K. Quadri, "A Software Reliability Growth Model with Two types of Learning", Proceedings of the 1st IEEE International Conference on Machine Intelligence Research and Advancement, SMVDU, Jammu, India, pp. 498–503, 2013

[6] J. Iqbal, N. Ahmad, and S.M.K. Quadri, "A software reliability growth model with two types of learning and a negligence factor," Image Information Processing (ICIIP), 2013 IEEE Second International Conference on , vol., no., pp.678,683, 9-11 Dec. 2013

[7] M. Ohba, "Software reliability analysis models," IBM Journal of Research Development, vol. 28, pp. 428–443, 1984.

[8] H. Pham, "Software reliability assessment: Imperfect debugging and multiple failure types in software development", 1993; EG&G-RAAM-10737, Idaho National Engineering Laboratory.

[9] P.K. Kapur, and S. Younes, "Modeling an imperfect debugging phenomenon in software reliability", Microelectronics and Reliability, 1996, Vol. 36, pp. 645-50.

[10] H.J. Shyur, "A stochastic software reliability model with imperfect debugging and change-point", J Syst Software, 2003, 66:135–41.

[11] K.C. Chiu, Y.S. Huang, and T.Z. Lee, "A study of software reliability growth from the perspective of learning effects", Reliability Engineering and System Safety, 2008, 93: 1410-1421.

[12] P. K. Kapur, H. Pham, S. Anand, K. Yadav, "A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation", IEEE Transactions on Reliability 60 (1) , 331–340, 2011.

[13] K.C. Chiu, "An improved model of software reliability growth under time-dependent learning effects," Quality and Reliability (ICQR), 2011 IEEE International Conference on , vol., no., pp.191,194, 14-17 Sept. 2011

[14] K.C. Chiu, "A discussion of software reliability growth models with time-varying learning effects",American Journal of Software engineering and applications. Vol. 2, No. 3, 2013, pp 92-104. doi: 10.11648/j.ajsea.20130203.12

[15] V. B. Singh, P. K. Kapur, and Mashaallah Basirzadeh. "Open Source Software Reliability Growth Model by Considering Change-Point." BVICAM's International Journal of Information Technology 4.

[16] Zhang X, Pham H. A software cost model with warranty cost, error removal times and risk costs. IIE Trans 1998;30:1135–42

[17] Hossain SA, Dahiya RC. Estimating the parameters of a non homogeneous Poisson-process model for software reliability. IEEETrans Reliab 1993;42:604–12.

[18] Pham H, Zhang X. NHPP software reliability and cost models with testing coverage. Eur J Oper Res 2003;145:445–54

[19] Bai CG, Hu QP, Xie M, Ng SH. Software failure prediction based on a Markov Bayesian network model. J Syst Software 2005;74:275–82

[20] Pham H. Software reliability and cost models- perspectives, comparison, and practice. Eur J Oper Res 2003;149:475–89

[21] Jeske DR, Zhang X. Some successful approaches to software reliability modeling in industry. J Syst Software 2005;74:85–99

[22] Anjum Mohd., Md. Asraful Haque, Nesar Ahmad. Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value, I.J. Information Technology and Computer Science, 2013, 02, pp.1-14, DOI: 10.5815/ijitcs.2013.02.01

[23] N. Ahmad, M.G.M. Khan, and L.S. Rafi, "Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging", International Journal of Computer Science and Network Security, 2011, Vol. 11 (1), pp. 161 – 171.

[24] N. Ahmad, M.G.M. Khan, and L.S. Rafi, "A Study of Testing-Effort Dependent Inflection S-Shaped Software Reliability Growth Models with Imperfect Debugging", International Journal of Quality and Reliability Management, 2010, Vol. 27 (1), pp. 89 – 110.

[25] P.K. Kapur, P.K. and S. Younes, "Modeling an imperfect debugging phenomenon in software reliability", Microelectronics and Reliability, 1996, Vol. 36, pp. 645-50.

[26] N. Ahmad, M.G.M. Khan, and L.S. Rafi, "Inflection S-shaped software reliability growth models with testing-effort functions," Proceedings of the VI International Symposium on Optimization and Statistics, Aligarh Muslim University, Aligarh, India, 29-31 December, 2008.

[27] N. Ahmad, M.U. Bokhari, S.M.K. Quadri, and M.G.M. Khan, "The exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy: a performance analysis", International Journal of Quality & Reliability Management, 2008, Vol. 25 No. 2, pp. 211-35.

[28] N. Ahmad, M.G.M. Khan, and L.S. Rafi, "Software Reliability Modeling Incorporating Log-Logistic Testing-Effort with Imperfect Debugging", in Proceedings of the International Conference on Modeling, Optimization and Computing (ICMOC-2010), Durgapur, India, Published by American Institute of Physics, 2010, pp. 651 – 657.

[29] J. Iqbal, S.M.K. Quadri, and N. Ahmad, "Software Reliability Modeling with Learning-Factor Based Fault-Detection Rate", Proceedings of the 1st International Conference on Recent Trends in Computer Science & Engineering" 8-9th February, 2014, Patna Bihar India (Narosa Publishing House)

[30] J. Iqbal, S.M.K. Quadri, and N. Ahmad, "An Imperfect- Debugging Model with Learning-Factor Based Fault-Detection Rate", Proceedings of the 8th International Conference on "Computing for Sustainable Global Development" (INDIACom-2014), Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA), 5th– 7thMarch 2014.

[31] Iqbal, Javaid. "Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations." *Cogent Engineering* 4.1 (2017): 1286739.