

# 基于高斯过程回归的软件可靠性模型

熊天骏<sup>1</sup>, 杨剑锋<sup>1,2\*</sup>, 王震<sup>1</sup>

<sup>1</sup>贵州大学数学与统计学院, 贵州 贵阳

<sup>2</sup>贵州理工学院大数据学院, 贵州 贵阳

收稿日期: 2023年6月2日; 录用日期: 2023年7月29日; 发布日期: 2023年8月3日

## 摘要

传统软件可靠性模型通常过于依赖假设条件, 难以适应复杂的实际情况。为此, 本文提出了一种基于高斯过程回归(Gaussian Process Regression, GPR)的软件可靠性非参数模型。该模型结合了机器学习和高斯过程核, 从失效数据中提取样本特征之间的相关关系。与传统可靠性模型相比, 本文提出的模型具有更广泛的应用效果。通过对两组真实数据进行对比分析, 结果显示本文提出的可靠性模型具有更好的拟合效果和预测能力。

## 关键词

软件可靠性模型, 高斯过程回归, 非参数模型, 机器学习

# A Software Reliability Model Based on Gaussian Process Regression

Tianjun Xiong<sup>1</sup>, Jianfeng Yang<sup>1,2\*</sup>, Zhen Wang<sup>1</sup>

<sup>1</sup>School of Mathematics and Statistics, Guizhou University, Guiyang Guizhou

<sup>2</sup>School of Big Data, Guizhou Institute of Technology, Guiyang Guizhou

Received: Jun. 2<sup>nd</sup>, 2023; accepted: Jul. 29<sup>th</sup>, 2023; published: Aug. 3<sup>rd</sup>, 2023

## Abstract

Traditional software reliability models often rely too much on assumptions and are difficult to adapt to complex practical situations. Therefore, this paper proposes a nonparametric model of software reliability based on Gaussian Process Regression (GPR). The model combines machine learning and Gaussian process kernels to extract correlations between sample features from failed data. Compared with the traditional reliability model, the model proposed in this paper has

\*通讯作者。

文章引用: 熊天骏, 杨剑锋, 王震. 基于高斯过程回归的软件可靠性模型[J]. 运筹与模糊学, 2023, 13(4): 2840-2849.

DOI: 10.12677/orf.2023.134284

a wider application effect. Through the comparative analysis of the two sets of real data, the results show that the reliability model proposed in this paper has better fitting effect and prediction ability.

## Keywords

Software Reliability Model, Gaussian Process Regression, Nonparametric Model, Machine Learning

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

软件可靠性是衡量软件系统在指定时间段内无故障地执行其预期功能的能力的指标，这是软件开发和维护的一个关键方面，因为软件故障可能会产生严重的后果，例如经济损失、声誉损害，甚至安全隐患，可靠的软件可以一致和准确地执行其预期功能，而不会造成任何意外或不良结果。软件可靠性受各种因素的影响，例如软件的复杂性、代码的质量、测试和验证过程以及操作环境。随着软件系统变得越来越复杂和互联，研究其可靠性变得越来越具有挑战性。

所以对软件可靠性的模型选择以及预测其未来增长趋势显得尤为重要，过去已有很多软件可靠性模型被提出，绝大多数模型是基于非齐次泊松过程(NHPP)类的软件可靠性模型。Yang 等[1]介绍了关于 NHPP 类开源软件可靠性增长模型的极大似然估计。Jelinski 等[2]介绍了软件可靠性的研究。Zhang 等[3]分析了影响软件可靠性的因素。最经典的模型为 G-O 模型，Jeske 等[4]提出了关于 G-O 模型的极大似然估计。Lavanya 等[5]研究了通过 aco 与 mle 的方法去估计 G-O 模型中的参数。同样用的很多的为 ISS 模型，对于 SRGM 没有透明的解释用于累积软件误差的方差估计，Lee 等[6]提出了利用 ISS 来建立具有置信区间的 SRGM。Erto 等[7]提出了一种新的广义下的 ISS 模型，它的特点是允许对每个故障函数的非单调故障率建模。Sagar 等[8]结合了威布尔分布方法和 ISS 模型。Pradhan 等[9]提出将测量工作量函数(TFF)作为时间变化故障率参与到广义 ISS 模型中。Wang 等[10]考虑将 ISS 模型引入开源软件可靠性模型。Shrivastava 等[11]通过考虑变化点前后的不同分布函数，开发了一个框架，将变化点纳入混合软件可靠性增长模型的开发中。Yang 等[12]提出一种考虑故障检测和纠错过程的屏蔽数据开源软件增长可靠性模型框架。

为了解决大规模复杂软件的可靠性预测和分析问题，软件可靠性模型需要具备处理多个参数的能力，贝叶斯网络表现出很强的适应涉及复杂变异因素的问题的能力，Bai 等[13]开发了一个扩展的马尔可夫贝叶斯网络，用于对具有操作配置文件的软件可靠性预测进行建模。Ruggeri 等[14]对贝叶斯软件可靠性建模进展进行了介绍。Wiper 等[15]分析了软件可靠性模型，其中软件度量形式的协变量信息可用性，使用神经网络回归，根据故障间隔时间或故障次数来估计模型中的故障率。Aktikin 等[16]开发了一个在软件可靠性方面调试不完善的模型，用贝叶斯分析，获取模型参数。

机器学习(ML)技术在软件可靠性预测中的应用已经显示出细致而显著的结果，Jaiswal 等[17]提出了使用 ML 技术进行软件可靠性预测，并根据选定的性能标准对其进行评估。Habtemariam 等[18]介绍了利用机器学习技术预测软件可靠性的系统文献综述。Mohanty 等[19]采用机器学习技术，来准确预测软件可靠性。

综上所述,本文的想法在于利用 GPR 模型来面对大规模复杂软件系统的可靠性预测和分析问题,增强预测的准确性,基于高斯过程回归的可靠性建模可以通过从历史数据中学习参数来计算复杂软件系统的初始可靠性。余下部分主要如下,第二章主要介绍了高斯过程回归的软件可靠性建模以及模型评价标准,第三章是案例分析,以及与 G-O 模型、ISS 模型的对比。第四章是结论。

## 2. 基于高斯过程回归的软件可靠性模型

### 2.1. 高斯过程回归的软件可靠性建模

高斯过程回归是来自于贝叶斯方法的一种非参数模型,高斯过程定义了一个先验函数,该函数在观察到先验分布中的一些值后转换为后验函数。本文只对一维高斯回归进行解释,对于训练集  $Z = (X_i, Y_i), i = 1, 2, 3, \dots, k$ 。  $X_i$  为输入数据,  $Y_i$  为对应的输出数据。目的是通过样本来学习,经过训练后,得出  $X_i$  和  $Y_i$  的函数关系,并用新的测试集  $X_i^*$  来预测  $Y_i^*$ 。

将训练集  $Z = (W_t, M_t), t = 1, 2, 3, \dots, s$ 。其中  $W$  为时间,  $M_t$  为  $t$  时刻的累计故障数,将  $Z$  代到(1)式,对于给定  $S$  组训练样本  $W = \{W_1, W_2, \dots, W_s\}$  有回归方程如下:

$$\begin{aligned} M &= f(W) + N(0, \sigma_n^2) \\ f(W) &\sim gp[0, k(W, W')] \end{aligned} \quad (1)$$

其中,  $W$  为学习样本,  $M$  为对应输出数据;  $gp[0, k(W, W')]$  是均值为 0、协方差为  $K$  的高斯过程;

由于高斯回归不需要先验知识,所以是从数据中训练后得到先验分布:

$$M \sim N(0, K(W, W') + \sigma_n^2 I) \quad (2)$$

而高斯回归中任意有限维的随机变量都服从高斯分布,所以对新输入的时间  $W^*$  以及对应预测故障数  $M^*$  也服从上述高斯分布,所以:

$$\begin{pmatrix} M \\ M^* \end{pmatrix} \sim N \left( 0, \begin{bmatrix} K(W, W) + \sigma_n^2 I & K(W, W^*) \\ K(W^*, W) & K(W^*, W^*) \end{bmatrix} \right) \quad (3)$$

再取边缘分布,有:

$$p(M^* | W, M, W^*, \sigma_n^2) = N(M^* | m(M^*), \text{cov}(M^*)) \quad (4)$$

其中,  $M^*$  的均值

$$m(M^*) = K(W^*, W) [K(W, W) + \sigma_n^2 I]^{-1} M \quad (5)$$

协方差

$$\text{cov}(M^*) = K(W^*, W^*) - K(W^*, W) [K(W, W) + \sigma_n^2 I]^{-1} K(W, W^*) \quad (6)$$

又上述可知,由于讲高斯过程中的均值预设为 0,所以完全由协方差函数决定,即可以通过(6)来算出预测值  $M^*$  以及其方差,所以协方差函数的选取尤为重要,本文选用的是径向基(RBF)核函数作协方差函数,RBF 是使用的最多的一种核函数,用于机器学习中的各种任务,如分类、回归和聚类。

### 2.2. 模型评价标准

本文将提出的 GPR 模型与 G-O 模型、ISS 模型进行对比。选用 RMSE 和 MAE 作为评价模型拟合程

度的指标:

均方根误差(RMSE): 预测值与实际值偏差的平方和与观察次数  $m$  比值的平方根:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^m (Y_i - Y_i^*)^2}{m}} \quad (7)$$

其中  $Y_i$  为实际值,  $Y_i^*$  为预测值。RMSE 更小说明效果更好。

平均绝对误差(MAE):

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |Y_i - Y_i^*| \quad (8)$$

其中  $Y_i$  为实际值,  $Y_i^*$  为预测值。 $m$  为回归数据的个数, MAE 越小表示模型的表现更好。

### 3. 案例分析

#### 3.1. 数据介绍

##### 3.1.1. 数据集 1

本文选取两组数据进行研究,各数据均来自用户缺陷跟踪系统。数据集 1 为 TOMCAT 服务器软件, Apache Tomcat, 通常简称为 Tomcat, 是由 Apache 软件基金会开发的开源 Web 服务器和 servlet 容器。它用于为 Java 应用程序提供服务, 并为运行基于 Java 的 Web 应用程序(如 JavaServer Pages (JSP)和 Java Servlets)提供环境。本文从缺陷系统中提取了 TOMCAT3-11 的 2000 年 8 月到 2023 年 5 月共计 274 个月的数据, 失效数据表如表 1 所示。其中第 1~193 月作为训练集, 第 194~274 月作为预测集。

**Table 1.** Accumulated failure data of TOMCAT3-11

**表 1.** TOMCAT3-11 的累计失效数据

时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数
1	3	53	2174	101	3054	149	4171	202	5233	250	5758
2	31	54	2192	102	3066	150	4198	203	5256	251	5775
3	57	55	2225	103	3074	151	4212	204	5266	252	5782
4	78	56	2245	104	3090	152	4245	205	5278	253	5787
5	113	57	2262	105	3111	153	4261	206	5288	254	5795
6	144	58	2284	106	3123	154	4283	207	5305	255	5796
7	174	59	2293	107	3145	155	4307	208	5322	256	5799
8	210	60	2312	108	3164	156	4365	209	5331	257	5806
9	244	61	2337	109	3174	157	4389	210	5346	258	5811
10	288	62	2357	110	3194	158	4401	211	5353	259	5814
11	324	63	2378	111	3207	159	4420	212	5361	260	5819
12	390	64	2399	112	3273	160	4438	213	5373	261	5825
13	436	65	2430	113	3294	161	4454	214	5383	262	5832
14	515	66	2454	114	3327	162	4491	215	5393	263	5836
15	590	67	2480	115	3356	163	4504	216	5401	264	5840

## Continued

16	659	68	2496	116	3389	164	4526	217	5413	265	5852
17	689	69	2512	117	3452	165	4559	218	5431	266	5856
18	739	70	2528	118	3465	166	4579	219	5443	267	5861
19	807	71	2549	119	3495	167	4602	220	5449	268	5867
20	859	72	2558	120	3516	168	4623	221	5463	269	5875
21	896	73	2571	121	3539	169	4639	222	5469	270	5881
22	932	74	2588	122	3585	170	4661	223	5480	271	5886
23	977	75	2605	123	3616	171	4689	224	5493	272	5900
24	1025	76	2624	124	3645	172	4711	225	5504	273	5906
25	1084	77	2634	125	3662	173	4729	226	5507	274	5907
26	1152	78	2647	126	3695	174	4747	227	5511		
27	1217	79	2671	127	3717	175	4775	228	5523		
28	1258	80	2691	128	3738	176	4808	229	5537		
29	1303	81	2709	129	3751	177	4827	230	5551		
30	1373	82	2738	130	3778	178	4841	231	5573		
31	1421	83	2761	131	3804	179	4850	232	5590		
32	1486	84	2777	132	3833	180	4867	233	5605		
33	1525	85	2797	133	3857	181	4877	234	5611		
34	1561	86	2817	134	3871	182	4928	235	5620		
35	1589	87	2850	135	3887	183	4949	236	5636		
36	1623	88	2869	136	3903	184	4973	237	5648		
37	1670	89	2876	137	3919	185	4982	238	5656		
38	1698	90	2887	138	3937	186	4991	239	5675		
39	1731	91	2901	139	3970	187	5003	240	5680		
40	1762	92	2915	140	3997	188	5031	241	5684		
41	1799	93	2926	141	4022	189	5044	242	5695		
42	1840	94	2939	142	4040	190	5058	243	5705		
43	1876	95	2957	143	4065	191	5066	244	5715		
44	1906	96	2978	144	4085	192	5086	245	5723		
49	2053	97	2992	145	4108	193	5102	246	5727		
50	2086	98	3008	146	4118	194	5119	247	5733		
51	2116	99	3031	147	4138	195	5131	248	5739		
52	2146	100	3042	148	4156	196	5155	249	5748		

## 3.1.2. 数据集 2

数据集 2 来源于 Apache HTTP，它是 Apache 软件基金会有一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，由于其多平台 and 安全性被广泛使用，是最流行的 Web 服务器端软件之一。本

文从缺陷系统中提取了 Apache httpd-2、Apache httpd-1.3、Apache httpd-test 的 2002 年 3 月到 2023 年 3 月共计 250 个月的累计失效数据。失效数据表如表 2 所示。其中第 1 个月~176 个月作为训练集，第 177 个月~第 250 个月作为预测集。

**Table 2.** Accumulated failure data of apache httpd  
**表 2.** Apache httpd 的累计失效数据

时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数	时间/月	累计失效数
1	13	49	1293	97	1979	145	2480	193	2820	241	3009
2	107	50	1315	98	1986	146	2486	194	2828	242	3013
3	167	51	1337	99	1994	147	2492	195	2834	243	3014
4	225	52	1357	100	2010	148	2498	196	2837	244	3017
5	279	53	1378	101	2021	149	2504	197	2848	245	3021
6	318	54	1394	102	2030	150	2512	198	2851	246	3022
7	365	55	1407	103	2036	151	2519	199	2856	247	3025
8	399	56	1424	104	2046	152	2532	200	2857	248	3025
9	435	57	1438	105	2074	153	2540	201	2861	249	3025
10	462	58	1458	106	2084	154	2548	202	2864	250	3028
11	511	59	1471	107	2095	155	2551	203	2876	241	3009
12	539	60	1479	108	2107	156	2562	204	2881	242	3013
13	569	61	1501	109	2117	157	2570	205	2884	243	3014
14	611	62	1519	110	2125	158	2576	206	2887	244	3017
15	634	63	1524	111	2133	159	2579	207	2890	245	3021
16	655	64	1535	112	2148	160	2586	208	2893	246	3022
17	689	65	1549	113	2161	161	2593	209	2896	247	3025
18	710	66	1587	114	2174	162	2601	210	2908	248	3025
19	722	67	1607	115	2181	163	2607	211	2910	249	3025
20	757	68	1622	116	2192	164	2611	212	2914	250	3028
21	783	69	1644	117	2203	165	2613	213	2920		
22	805	70	1660	118	2209	166	2621	214	2921		
23	824	71	1669	119	2216	167	2639	215	2928		
24	841	72	1684	120	2232	168	2649	216	2934		
25	876	73	1699	121	2260	169	2657	217	2939		
26	908	74	1718	122	2280	170	2665	218	2943		
27	927	75	1731	123	2291	171	2673	219	2947		
28	949	76	1744	124	2297	172	2678	220	2948		
29	973	77	1763	125	2301	173	2688	221	2949		
30	988	78	1773	126	2309	174	2700	222	2951		

Continued

31	1015	79	1785	127	2319	175	2704	223	2952
32	1027	80	1799	128	2326	176	2709	224	2956
33	1039	81	1812	129	2334	177	2716	225	2958
34	1054	82	1822	130	2346	178	2722	226	2961
35	1066	83	1831	131	2356	179	2730	227	2963
36	1083	84	1839	132	2366	180	2733	228	2964
37	1098	85	1847	133	2375	181	2739	229	2965
38	1113	86	1857	134	2382	182	2741	230	2966
39	1125	87	1868	135	2388	183	2746	231	2967
40	1148	88	1879	136	2394	184	2759	232	2969
41	1156	89	1885	137	2404	185	2766	233	2971
42	1167	90	1897	138	2420	186	2769	234	2976
43	1181	91	1903	139	2425	187	2778	235	2986
44	1201	92	1921	140	2432	188	2785	236	2993
45	1215	93	1933	141	2439	189	2787	237	2996
46	1238	94	1949	142	2446	190	2793	238	2999
47	1256	95	1961	143	2459	191	2800	239	3002
48	1273	96	1970	144	2466	192	2804	240	3007

### 3.2. 模型性能比较分析

#### 3.2.1. 数据集 1 性能对比

通过使用 2.1 节提出的模型, 并使用 3.1 的数据集进行训练, 我们得到了 G-O 模型以及 ISS 模型的参数估计结果和模型比较结果, 如表 3 所示, 从表中可以得出, 在训练集方面, GPR 的 RMSE 要比 G-O 模型以及 ISS 模型的 RMSE 要低, 并且 GPR 的 MAE 也比 G-O 模型以及 ISS 模型的 MAE 要低, 说明 GPR 模型拟合效果较好, 其次是 ISS 模型, 最差的是 G-O 模型。在测试集方面, GPR 的 RMSE 以及 MAE 要远低于 G-O 模型以及 ISS 模型的 RMSE 和 MAE, 说明 GPR 模型的预测效果更好, 其次是 ISS 模型, 预测效果最差是 G-O 模型。图 1 是数据集 1 的效果图, 训练集用虚线表示, 测试集用实线表示。

**Table 3.** Parameter estimation and model comparison results for training set 1

**表 3.** 训练集 1 的参数估计和模型比较结果

数据集	模型	参数估计结果			RMSE_train	MAE_train	RMSE_pd	MAE_pd
		<i>a</i>	<i>b</i>	<i>c</i>				
1	G-O	7251.8054	0.0058		145.6371	125.2658	171.1652	168.8147
	ISS	11487.0952	2.4399	0.000024	135.5570	117.0393	118.7530	105.7319
	GPR				9.9031	7.0323	15.3514	13.3825

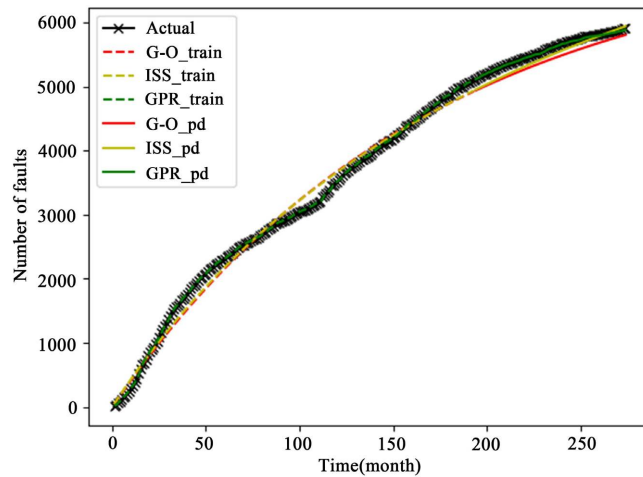


Figure 1. Renderings of dataset 1

图 1. 数据集 1 的效果图

### 3.2.2. 数据集 2 性能对比

如表 4 所示,在训练集中,GPR 的 RMSE 和 MAE 都要低于 G-O 模型以及 ISS 模型的 RMSE 和 MAE,说明 GPR 模型的拟合效果更好,在测试集中,GPR 的 RMSE 和 MAE 也要远远低于 G-O 模型以及 ISS 模型的 RMSE 和 MAE,说明 GPR 模型的预测效果更好,ISS 模型预测效果也不错,最差的是 G-O 模型。图 2 是数据 2 的效果图。

Table 4. Parameter estimation and model comparison results for training set 2

表 4. 训练集 2 的参数估计和模型比较结果

数据集	模型	参数估计结果			RMSE_train	MAE_train	RMSE_pd	MAE_pd
		$a$	$b$	$c$				
2	G-O	3028.4343	0.0115		63.1556	48.8927	140.9678	138.2626
	ISS	4319.6777	0.0001	-0.9876	47.4863	37.8487	54.0586	52.2695
	GPR				6.8872	4.4092	3.6197	2.5909

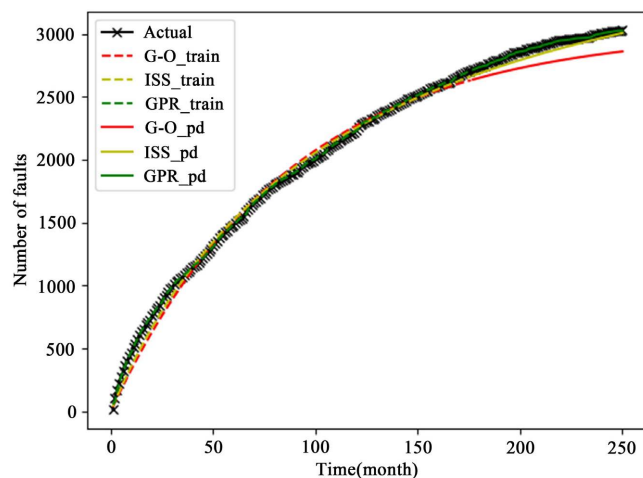


Figure 2. Renderings of dataset 2

图 2. 数据集 2 的效果图



## 4. 结论

随着互联网的发展, 软件的规模越来越大, 传统可靠性模型预测方法去预测越来越难, 拟合度也没有太好, 基于现代软件的规模, 利用基于贝叶斯框架的高斯过程回归去拟合累计故障数据, 建立预测的模型, 并与其他两种传统可靠性模型进行了比较, 结果表明, 本文提出的 GPR 在准确性和效率方面优于它们, 是一种不错的可靠性预测方法, 为软件可靠性预测领域提供了有价值的贡献。

## 基金项目

国家自然科学基金资助项目(71901078); 贵州省电力大数据重点实验室(黔科合计 Z 字[2015] 4001)。

## 参考文献

- [1] 陈静, 杨剑锋, 王喜宾, 等. NHPP 类开源软件可靠性增长模型的极大似然估计[J]. 广西大学学报(自然科学版), 2022, 47(1): 174-184.
- [2] Jelinski, Z. and Moranda, P. (1972) Software Reliability Research. In: Freiberger, W., Ed., *Statistical Computer Performance Evaluation*, Academic Press, Cambridge, 465-484. <https://doi.org/10.1016/B978-0-12-266950-7.50028-1>
- [3] Zhang, X. and Pham, H. (2000) An Analysis of Factors Affecting Software Reliability. *Journal of Systems and Software*, **50**, 43-56. [https://doi.org/10.1016/S0164-1212\(99\)00075-8](https://doi.org/10.1016/S0164-1212(99)00075-8)
- [4] Jeske, D.R. and Pham, H. (2001) On the Maximum Likelihood Estimates for the Goel-Okumoto Software Reliability Model. *The American Statistician*, **55**, 219-222. <https://doi.org/10.1198/000313001317098211>
- [5] Lavanya, G., Neeraja, K., Basha, S.A., et al. (2017) Parameter Estimation of Goel-Okumoto Model by Comparing ACO with MLE Method. *International Research Journal of Engineering and Technology*, **4**, 1605-1615.
- [6] Lee, T.-Q., Fang, C.-C. and Yeh, C.-W. (2013) Confidence Interval Estimation of Software Reliability Growth Models Based on Ohba's Inflection S-Shaped Model. *Journal of Industrial and Intelligent Information*, **1**, 196-200. <https://doi.org/10.12720/jiii.1.4.196-200>
- [7] Erto, P., Giorgio, M. and Lepore, A. (2020) The Generalized Inflection S-Shaped Software Reliability Growth Model. *IEEE Transactions on Reliability*, **69**, 228-244. <https://doi.org/10.1109/TR.2018.2869466>
- [8] Sagar, B.B., Saket, R.K. and Gurmit, S.C. (2016) Exponentiated Weibull Distribution Approach Based Inflection S-Shaped Software Reliability Growth Model. *Ain Shams Engineering Journal*, **7**, 973-991. <https://doi.org/10.1016/j.asej.2015.05.009>
- [9] Pradhan, V., Kumar, A. and Dhar, J. (2022) Enhanced Growth Model of Software Reliability with Generalized Inflection S-Shaped Testing-Effort Function. *Journal of Interdisciplinary Mathematics*, **25**, 137-153. <https://doi.org/10.1080/09720502.2021.2006329>
- [10] Wang, J. and Zhang, C. (2022) Reliability Model of Open Source Software Considering Fault Introduction with Generalized Inflection S-Shaped Distribution. *SN Applied Sciences*, **4**, Article No. 244. <https://doi.org/10.1007/s42452-022-05125-6>
- [11] Shrivastava, A.K. and Sharma, R. (2022) Developing a Hybrid Software Reliability Growth Model. *International Journal of Quality & Reliability Management*, **39**, 1209-1225. <https://doi.org/10.1108/IJQRM-02-2021-0039>
- [12] Yang, J., Zhao, M. and Chen, J. (2022) ELS Algorithm for Estimating Open Source Software Reliability with Masked Data Considering both Fault Detection and Correction Processes. *Communications in Statistics-Theory and Methods*, **51**, 6792-6817. <https://doi.org/10.1080/03610926.2020.1866610>
- [13] Bai, C.-G. (2005) Bayesian Network Based Software Reliability Prediction with an Operational Profile. *Journal of Systems and Software*, **77**, 103-112. <https://doi.org/10.1016/j.jss.2004.11.034>
- [14] Ruggeri, F., Soyer, R. and Imati, C. (2008) Advances in Bayesian Software Reliability Modelling. In: Bedford, T., et al., Eds., *Advances in Mathematical Modelling for Reliability*, IOS Press, Amsterdam, 149-157.
- [15] Wiper, M.P., Palacios, A.P. and Marin, J.M. (2012) Bayesian Software Reliability Prediction Using Software Metrics Information. *Quality Technology & Quantitative Management*, **9**, 35-44. <https://doi.org/10.1080/16843703.2012.11673276>
- [16] Aktekin, T. and Caglar, T. (2013) Imperfect Debugging in Software Reliability: A Bayesian Approach. *European Journal of Operational Research*, **227**, 112-121. <https://doi.org/10.1016/j.ejor.2012.11.056>
- [17] Jaiswal, A. and Malhotra, R. (2018) Software Reliability Prediction Using Machine Learning Techniques. *International Journal of System Assurance Engineering and Management*, **9**, 230-244. <https://doi.org/10.1007/s13198-016-0543-y>

- 
- [18] Habtemariam, G.M., Mohapatra, S.K., Seid, H.W., *et al.* (2022) A Systematic Literature Review of Predicting Software Reliability Using Machine Learning Techniques. In: Khari, M., Mishra, D.B., Acharya, B. and Crespo, R.G., Eds., *Optimization of Automated Software Testing Using Meta-Heuristic Techniques*, Springer, Berlin, 77-90.  
[https://doi.org/10.1007/978-3-031-07297-0\\_6](https://doi.org/10.1007/978-3-031-07297-0_6)
- [19] Mohanty, R., Ravi, V. and Patra, M.R. (2010) Application of Machine Learning Techniques to Predict Software Reliability. *International Journal of Applied Evolutionary Computation (IJAE)*, **1**, 70-86.  
<https://doi.org/10.4018/jaec.2010070104>