

基于随机微分方程和广义拐点S型故障检测率的银行软件可靠性研究

王震¹, 杨剑锋^{1,2*}, 熊天骏¹

¹贵州大学数学与统计学院, 贵州 贵阳

²南宁师范大学数学与统计学院, 广西 南宁

收稿日期: 2024年9月12日; 录用日期: 2024年10月16日; 发布日期: 2024年11月20日

摘要

在数字经济时代, 随着电子商务快速发展, 银行软件系统的规模显著扩大, 软件故障跟踪系统中故障的报告存在随机性、不规律性以及各种不确定因素, 因此故障检测过程可以看作是一个随机过程。在软件可靠性领域, 随机微分方程常用于描述软件故障的发生过程及其可靠性的动态变化, 在故障检测率方面, 广义拐点S型故障检测率往往具有更强的灵活性, 因此本文提出一种基于随机微分方程和广义拐点S型故障检测率的软件可靠性模型, 并使用极大似然估计方法对模型的参数进行估计。最后在A银行测试中心数据集上进行验证, 实验结果表明本文提出模型对实验数据拟合效果最好。因此该模型相较于其他经典软件可靠性模型具有优越性。

关键词

随机微分方程, 软件可靠性模型, 银行软件, 电子商务

Research on the Reliability of Banking Software Based on Stochastic Differential Equations and Generalized Inflection Point S-Type Fault Detection Rate

Zhen Wang¹, Jianfeng Yang^{1,2*}, Tianjun Xiong¹

¹School of Mathematics and Statistics, Guizhou University, Guiyang Guizhou

²School of Mathematics and Statistics, Nanning Normal University, Nanning Guangxi

Received: Sep. 12th, 2024; accepted: Oct. 16th, 2024; published: Nov. 20th, 2024

*通讯作者。

文章引用: 王震, 杨剑锋, 熊天骏. 基于随机微分方程和广义拐点 S 型故障检测率的银行软件可靠性研究[J]. 电子商务评论, 2024, 13(4): 3719-3727. DOI: 10.12677/ec.2024.1341575

Abstract

In the era of digital economy, with the rapid development of e-commerce, the scale of banking software systems has expanded significantly, and the fault reporting in the software fault tracking system has randomness, irregularity and various uncertain factors, so the fault detection process can be regarded as a random process. In the field of software reliability, stochastic differential equations are often used to describe the occurrence process of software failures and the dynamic changes of their reliability, and the generalized inflection point S-type fault detection rate is often more flexible in terms of fault detection rate, so this paper proposes a software reliability model based on stochastic differential equations and generalized inflection point S-type fault detection rate, and uses the maximum likelihood estimation method to estimate the parameters of the model. Finally, it is verified on the data set of A Bank test center, and the experimental results show that the proposed model has the best fitting effect on the experimental data. Therefore, this model has advantages over other classical software reliability models.

Keywords

Stochastic Differential Equations, Software Reliability Models, Banking Software, Electronic Commerce

Copyright © 2024 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

随着传统金融行业面向数字化转型的迫切需求, 银行的各种业务交易和各项金融服务由软件系统完成, 其软件系统的建设工作均被信息部门承接。银行软件的核心是处理资金交易, 任何系统故障或安全漏洞都会导致资金损失, 甚至引发大规模的金融风险。可靠的软件可以确保交易的准确性和实时性, 避免因系统问题带来的资金错误或丢失。一旦软件发生故障, 就会对银行业务造成影响甚至可能造成巨大损失。例如: 2012 年, 英国皇家银行因软件更新失败导致了广泛的系统故障, 影响了数百万客户的日常银行业务。问题源于系统升级过程中一项操作出现错误, 导致银行的核心账务系统无法正常处理交易。2018 年, 英国 TSB 银行在将其 IT 系统从母公司西班牙萨班德银行的平台迁移到自主平台时, 发生了严重的技术故障。迁移过程中的错误导致银行系统瘫痪, 客户无法访问账户长达数周。如今已经有大量研究人员对软件可靠性模型进行研究, 其中最常用的一类是 NHPP (Non-homogeneous Poisson process) 类软件可靠性模型, 该模型的特点是假设累计故障数服从非齐次泊松过程。原始的 NHPP 类模型是 G-O 模型, 其后相继提出了 Delayed S-shaped、Inflection S-shaped 以及 Yamada 等经典模型。

Wang 等[1]研究了软件测试人员在排除故障时也有学习的现象, 当检测到的故障排除后, 引入的故障数量逐渐减少, 建立了一种故障引入的趋势呈递减趋势的软件可靠性模型。Heliyon 等[2]提出一种基于动态加权 NHPP 模型的基于增强邻域的粒子群优化(NPSO)技术的新方法, 用于带有附加故障引入参数的软件可靠性分析。Li 等[3]提出一种将 S 型测试工作量函数纳入具有不完善调试的 NHPP 软件可靠性模型。Li 等[4]将不完善的调试和运行环境的不确定性及其对故障检测率的影响涵盖到基于非齐次泊松过程(NHPP)的软件可靠性评估中。Jain [5]提出一种引入测试工作量(TEF)和故障减少因子(FRF)和不完善的调

试软件可靠性增长模型。Song [6]考虑软件测试受环境的影响,提出一种具有受操作环境影响的威布尔故障检测率函数的软件可靠性模型。后续研究学者还利用随机微分方程来建立软件可靠性模型。Chaudhary [7]在故障检测率中融入了变化点概念,提出了一种基于随机微分方程(SDE)的软件可靠性增长模型。Chatterjee [8]考虑软件可靠性受软件故障的严重程度以及各故障被修复的优先级的影响,建立了一种基于随机微分方程的软件可靠性模型,将故障检测与故障发生的严重程度联系在一起,并将纠正故障的过程与缺陷修复的优先级联系在一起。Yang 等[9]考虑了具有多个子模块的软件系统的复杂性,引入屏蔽数据,提出一种基于随机微分方程(SDE)的多组件软件可靠性加性模型。Singh 等[10]考虑软件故障修复、功能增强和添加新功能等情况,给故障排除率带来了不确定性,使用三维维纳过程定义了三种波动类型,提出了一个统一的模型框架。Kapur [11]考虑到大规模软件每次调试检测到并消除的故障数量的变化与测试阶段开始时的初始故障内容相比变得足够小,提出一种基于随机微分方程的广义 Erlang 模型。Fang [12]等采用随机微分方程对软件故障检测过程进行置信区间估计,为软件开发人员和测试人员进行软件开发和软件质量控制提供了有益的信息。本文考虑了软件故障跟踪系统中故障报告存在随机性、不规律性以及各种不确定因素的影响,利用随机微分方程建立软件可靠性模型,并且为了更好地适应银行软件故障特点,采用广义拐点 S 型分布作为故障检测率函数。

2. 随机微分方程软件可靠性模型框架

2.1. 模型假设

随着软件系统规模不断扩大,故障检测过程充满不确定性。针对这种不确定性,将系统的故障视为一种随机过程,利用随机微分方程对系统可靠性进行建模。

基于随机微分方程软件可靠性模型假设如下:

- 1) 故障检测过程可以看作是一个随机过程。
- 2) 软件系统中残留的故障数量随着测试时间逐渐减少,并且检测到的故障去除时不会引入新的故障。
- 3) 在故障检测过程中,故障检测率随测试时间不规则变化。
- 4) 系统剩余缺陷引起软件失效。

记 $N(t)$ 为 $(0, t]$ 时间段内检测到的累积故障数, $\{N(t), t \geq 0\}$ 是一个独立增量过程;由假设可以得到微分方程:

$$\frac{dN(t)}{dt} = r(t)[a - N(t)] \quad (1)$$

$$r(t) = b(t) + \delta\gamma(t) \quad (2)$$

其中, a 表示软件系统中预期的总故障数, $b(t)$ 为故障检测率函数,表示时间 t 时每个故障被检测的概率。 $r(t)$ 含有不规则波动,其中 δ 表示不规则波动的正常量, $\gamma(t)$ 为标准高斯白噪声,并且有下列性质,

$$E(\delta\gamma(t)) = 0$$

$$\text{Var}(\delta\gamma(t)) = \delta^2$$

将式(1)代入(2)可以得到:

$$\frac{dN(t)}{dt} = (b(t) + \delta\gamma(t))[a - N(t)] \quad (3)$$

式(3)可以被扩展成下列的伊藤(Ito-type)类随机微分方程,

$$dN(t) = \left[b(t) - \frac{1}{2} \delta^2 \right] (a - N(t)) dt + \delta (a - N(t)) dW(t) \quad (4)$$

其中 $W(t)$ 表示布朗运动或者是一维维纳过程。

2.2. 基于随机微分方程和广义拐点 S 型故障检测率的软件可靠性模型

为了适应银行软件测试数据, 本文采用广义拐点 S 型故障检测率函数作为 $b(t)$, 该函数允许对每个故障函数的非单调故障检测率进行建模, 是一种非常灵活的故障检测率函数。即:

$$b(t) = \frac{kct^{c-1}}{1 + be^{-kt^c}}; t \geq 0, k, c > 0, b \geq 0 \quad (5)$$

其中, k 、 c 和 b 分别是尺度参数、形状参数和拐点参数, 用来表示软件测试中的故障检测行为。根据(3~5)即可得到随机微分方程:

$$dN(t) = \left[\frac{kct^{c-1}}{1 + be^{-kt^c}} - \frac{1}{2} \delta^2 \right] (a - N(t)) dt + \delta (a - N(t)) dW(t) \quad (6)$$

通过伊藤公式求解 $N(t)$:

$$N(t) = a \left[1 - \frac{1 + b}{1 + be^{-kt^c}} e^{-kt^c - \delta W(t)} \right] \quad (7)$$

得到本文提出的 GISS_SDE 模型, 其期望检测出的故障数量 $m(t)$ 可以表示为:

$$m(t) = a \left(1 - \frac{(1 + b) \exp\left(-kt^c + \frac{\delta^2 t}{2}\right)}{1 + b \exp(-kt^c)} \right) \quad (8)$$

2.3. 模型的参数的极大似然估计

记 (n_i, t_i) ($i = 0, 1, 2, \dots, N; 0 = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_N$) 表示 N 个观察到的系统软件故障数据对, n_i 表示在时间区间 $[0, t_i]$ 内检测到的故障数量, $N(t_i)$ 的联合分布函数可以表示为:

$$F(t_1, n_1; t_2, n_2; \dots; t_N, n_N) = P(N(t_1) \leq n_1, \dots, N(t_N) \leq n_N | N(t_0) = 0) \quad (9)$$

其中, $N(t_i)$ 表示系统软件在 t_i 时刻累计检测出的故障数量。根据式(7)和(9), $N(t_i)$ 的概率分布为:

$$\begin{aligned} & P[N(t_i) \leq n_i | N(t_{i-1}) = n_{i-1}] \\ &= \phi \left[\frac{\log \left[\frac{a - n_{i-1}}{a - n_i} \right] - \left[\int_0^{t_i} b(s) ds - \int_0^{t_{i-1}} b(s) ds \right]}{\delta \sqrt{t_i - t_{i-1}}} \right] \\ &= \phi \left[\frac{\log \left(\frac{a - n_{i-1}}{a - n_i} \right) - k(t_i^c - t_{i-1}^c) + \log \left(\frac{1 + be^{-kt_{i-1}^c}}{1 + be^{-kt_i^c}} \right)}{\delta \sqrt{t_i - t_{i-1}}} \right] \end{aligned} \quad (10)$$

构造对数似然函数:

$$\begin{aligned}
L &= \log \left(\prod_{i=1}^N \frac{d}{dn_i} P(N(t_i) \leq n_i | N(t_{i-1}) = n_{i-1}) \right) \\
&= \sum_{i=1}^N \left[-\log(\delta(a - n_i)) - \frac{1}{2} \log(2\pi(t_i - t_{i-1})) \right] \\
&\quad - \sum_{i=1}^N \frac{1}{2\delta^2(t_i - t_{i-1})} \times \left[\log\left(\frac{a - n_{i-1}}{a - n_i}\right) - k(t_i^c - t_{i-1}^c) + \log\left(\frac{1 + be^{-kt_i^c}}{1 + be^{-kt_{i-1}^c}}\right) \right]^2
\end{aligned} \tag{11}$$

本文提出模型的参数 a , b , c , k 和 δ 的最大似然估计值 $\hat{a}$, $\hat{b}$, $\hat{c}$, $\hat{k}$, $\hat{\delta}$ 可以通过解以下方程得到。

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial b} = \frac{\partial L}{\partial c} = \frac{\partial L}{\partial k} = \frac{\partial L}{\partial \delta} = 0 \tag{12}$$

2.4. 模型评价指标

均方误差(MSE): 预测值与实际值偏差的平方和与观察次数 N 的比值:

$$MSE = \frac{1}{N} \sum_i [m(t_i) - n_i]^2 \tag{13}$$

其中 $m(t_i)$ 为模型的预测值, n_i 为 t_i 时刻真实故障数。MSE 值越低, 表明拟合误差越小。

3. 数值案例

为了充分、有效地评估提出的改进模型的拟合和预测性能, 本节使用了经典 NHPP 模型: GO 模型、DSS模型、ISS 模型, GISS 模型以及本文提出的 GISS_SDE 模型对计算结果进行对比分析, 见表 1。

Table 1. Model summary table

表 1. 模型汇总表

模型序号	模型名称	模型的均值函数
Model1	GO	$m(t) = a(1 - \exp(-kt))$
Model2	ISS	$m(t) = a \left(1 - \frac{1+b}{1+b\exp(-kt)} \exp(-kt) \right)$
Model3	DSS	$m(t) = a(1 - (1+kt)\exp(-kt))$
Model4	GISS	$m(t) = a \left(1 - \frac{(1+b)\exp(-kt^c)}{1+b\exp(-kt^c)} \right)$
Model5	GISS_SDE (本文提出的模型)	$m(t) = a \left(1 - \frac{(1+b)\exp\left(-kt^c + \frac{\delta^2 t}{2}\right)}{1+b\exp(-kt^c)} \right)$

3.1. 数据说明

本文数据来自 MISYS 资金交易系统项目，MISYS 资金交易系统是 A 银行外购的一个资金业务中后台系统。该项目测试对象包含在此升级过程中所涉及或关联到的所有系统。系统包括：资金产品管理系统、数据仓库系统、主机业务系统(境内)、资金业务管理系统(境内)等。选取 2020 年 6 月 29 日到 2020 年 9 月 2 日检测到的故障数(共 33 组数据)，累计检测故障 133 个。

3.2. 参数估计结果

使用表 2 中 A 银行 MISYS 资金交易系统项目失效数据对经典模型和本文模型进行参数估计，参数估计结果及评价指标见表 3。

Table 2. Bank A's MISYS fund trading system project failure data
表 2. A 银行 MISYS 资金交易系统项目失效数据

时间	累计故障数	时间	累计故障数	时间	累计故障数
1	0	14	70	27	103
2	3	15	71	28	105
3	5	16	83	29	107
4	20	17	76	30	109
5	28	18	80	31	110
6	33	19	82	32	111
7	37	20	84	33	113
8	42	21	86		
9	46	22	89		
10	51	23	93		
11	52	24	84		
12	57	25	96		
13	65	26	102		

Table 3. Model summary table
表 3. 模型汇总表

模型名称	参数估计值	MSE
GO	$a = 159.371$	32.816
	$k = 0.08154$	
ISS	$a = 199.5048$	21.25524
	$b = -0.8601$	
	$k = 0.0108$	

续表

DSS	$a = 141.4787$	130.1294
	$k = 0.2212$	
GISS	$a = 204.317$	21.24545
	$b = 0.9210$	
	$c = 0.6356$	
	$k = 0.2068$	
GISS_SDE 本文提出的模型	$a = 225$	21.21
	$b = 0.9771$	
	$c = 0.6018$	
	$\delta = 0.00785$	
	$k = 0.2038$	

3.3. 模型性能对比分析

从图 1 和表 3 可以看出经典非齐次泊松过程模型里 GISS 模型效果最好，MSE 值为 21.245，其次是 ISS 模型，MSE 值为 21.2524，DSS 模型效果最差，MSE 值为 130.1294，针对 GISS 模型来说，本文改进后的 GISS_SDE 模型效果要优于 GISS 模型，其 MSE 值为 21.21。

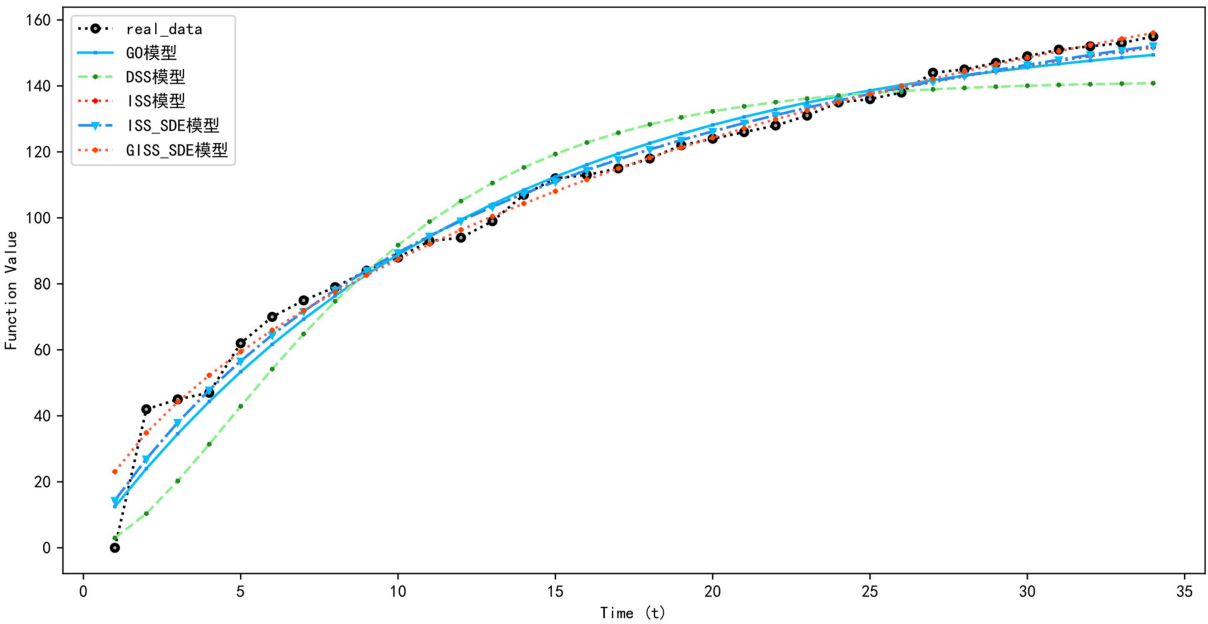


Figure 1. Fitting diagram of the cumulative number of faults of each model

图 1. 各模型累计故障数拟合图

图 2 为相对误差拟合对比图，通过绘制各个模型的相对误差曲线(Relative error curve)，GISS_SDE 模

型随着时间的增长,其波动程度最小,并最终趋近于 0,表明 GISS_SDE 模型对数据拟合的适应性最强。因此本文提出的模型更适合评估所选银行软件的可靠性。

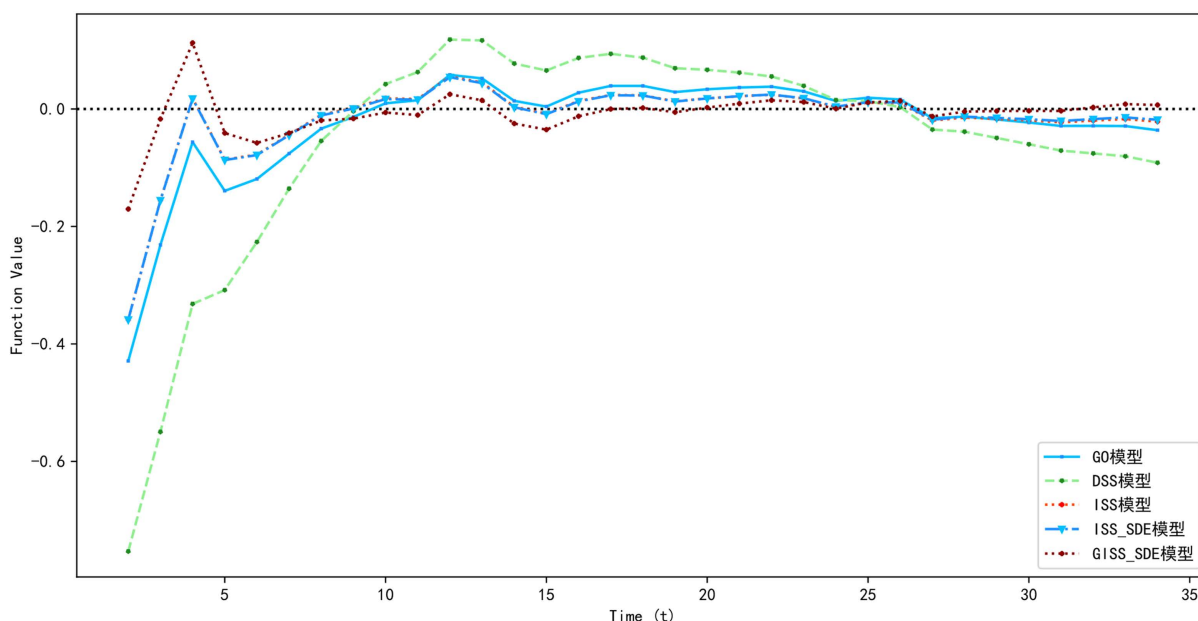


Figure 2. Comparison chart of the relative error fitting of each model

图 2. 各模型相对误差拟合对比图

4. 结论

在数字经济时代,软件的可靠性对银行的影响深远且广泛,直接关乎银行的运营效率、客户体验、业务连续性和市场竞争力。然而银行软件系统规模的显著扩大,导致软件测试故障的报告存在随机性、不规律性以及各种不确定因素,这些因素往往会影响到银行软件可靠性评估。本文基于随机微分方程将广义拐点 S 型分布作为故障检测率,提出一种新的软件可靠性模型,并将其应用到 A 银行 MISYS 资金交易系统项目中。通过绘制累计故障数拟合图和各模型相对误差对比图得出结论,本文提出的模型相较于其他经典非齐次泊松过程类软件可靠性模型具有更好的拟合效果,为评估银行软件可靠性提供了有价值的贡献。

参考文献

- [1] Wang, J., Zhang, C. and Yang, J. (2022) Software Reliability Model of Open Source Software Based on the Decreasing Trend of Fault Introduction. *PLOS ONE*, **17**, e0267171. <https://doi.org/10.1371/journal.pone.0267171>
- [2] Rani, P. and Mahapatra, G.S. (2019) A Novel Approach of NPSO on Dynamic Weighted NHPP Model for Software Reliability Analysis with Additional Fault Introduction Parameter. *Heliyon*, **5**, e02082. <https://doi.org/10.1016/j.heliyon.2019.e02082>
- [3] Li, Q., Li, H. and Lu, M. (2015) Incorporating S-Shaped Testing-Effort Functions into NHPP Software Reliability Model with Imperfect Debugging. *Journal of Systems Engineering and Electronics*, **26**, 190-207. <https://doi.org/10.1109/jsee.2015.00024>
- [4] Li, Q. and Pham, H. (2019) A Generalized Software Reliability Growth Model with Consideration of the Uncertainty of Operating Environments. *IEEE Access*, **7**, 84253-84267. <https://doi.org/10.1109/access.2019.2924084>
- [5] Jain, M., Manjula, T. and Gulati, T.R. (2014) Imperfect Debugging Study of SRGM with Fault Reduction Factor and Multiple Change Point. *International Journal of Mathematics in Operational Research*, **6**, 155-175. <https://doi.org/10.1504/ijmor.2014.059526>

-
- [6] Song, K., Chang, I. and Pham, H. (2017) A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments. *Applied Sciences*, 7, Article 983. <https://doi.org/10.3390/app7100983>
 - [7] Chaudhary, K., Kapur, P.K., Kumar, P. and Kumar, V. (2024) Stochastic Software Reliability Growth Modelling with Fault Introduction and Change Point. *International Journal of System Assurance Engineering and Management*. <https://doi.org/10.1007/s13198-024-02406-8>
 - [8] Chatterjee, S., Chaudhuri, B. and Bhar, C. (2020) Optimal Release Time Determination via Fuzzy Goal Programming Approach for SDE-Based Software Reliability Growth Model. *Soft Computing*, 25, 3545-3564. <https://doi.org/10.1007/s00500-020-05385-7>
 - [9] Yang, J., Ding, M., He, M., Zheng, Z. and Yang, N. (2024) SDE-Based Software Reliability Additive Models with Masked Data Using ELS Algorithm. *Journal of King Saud University-Computer and Information Sciences*, 36, Article 101978. <https://doi.org/10.1016/j.jksuci.2024.101978>
 - [10] Deepika, Singh, O., Anand, A. and Singh, J. (2020) SDE Based Unified Scheme for Developing Entropy Prediction Models for OSS. *International Journal of Mathematical, Engineering and Management Sciences*, 6, 207-222. <https://doi.org/10.33889/ijmems.2021.6.1.013>
 - [11] Kapur, P.K., Anand, S., Yamada, S. and Yadavalli, V.S.S. (2009) Stochastic Differential Equation-Based Flexible Software Reliability Growth Model. *Mathematical Problems in Engineering*, 2009, Article 581383. <https://doi.org/10.1155/2009/581383>
 - [12] Fang, C. and Yeh, C. (2015) Effective Confidence Interval Estimation of Fault-Detection Process of Software Reliability Growth Models. *International Journal of Systems Science*, 47, 2878-2892. <https://doi.org/10.1080/00207721.2015.1036474>