

# 基于向量自回归模型的软件可靠性评估与预测方法

严可<sup>1</sup>, 杨剑锋<sup>1,2\*</sup>, 冯宝凤<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: 2022年9月17日; 录用日期: 2022年10月8日; 发布日期: 2022年10月17日

## 摘要

针对目前大多数软件可靠性模型未考虑故障相关性和参数估计求解较难的问题, 本文提出了一种基于向量自回归模型的软件可靠性评估与预测方法, 该模型考虑了软件各组件之间的相关性并利用组件的故障数据来进行建模。所提出的软件可靠性预测模型被应用于两个真实的软件故障数据集, 实验结果表明, 本文提出的基于向量自回归的软件可靠性预测模型效果更优。

## 关键词

软件可靠性, 参数估计, 向量自回归模型, 故障相关性

# Software Reliability Assessment and Prediction Method Based on Vector Autoregressive Model

Ke Yan<sup>1</sup>, Jianfeng Yang<sup>1,2\*</sup>, Baofeng Feng<sup>1</sup>

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

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

Received: Sep. 17<sup>th</sup>, 2022; accepted: Oct. 8<sup>th</sup>, 2022; published: Oct. 17<sup>th</sup>, 2022

## Abstract

Aiming at the problem that most of the current software reliability models do not consider fault dependent and parameter estimation, this paper proposed a software reliability assessment and

\*通讯作者。

prediction method based on vector autoregressive model, which considers the correlation between software components and uses the component fault data for modeling. The proposed software reliability prediction model is applied to two real software project data sets. The experimental results show that the software reliability prediction model based on vector autoregression proposed in this paper is more effective.

## Keywords

Software Reliability, Parameter Estimation, Vector Autoregressive Model, Fault Correlation

Copyright © 2022 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

软件可靠性工程是软件开发的一个重要部分,在该领域的研究也越来越多。软件可靠性模型在软件可靠性工程概念领域发挥着至关重要的作用;它可以在测试阶段的某个阶段为软件开发和质量保证团队提供相当有价值的信息。软件可靠性有助于降低公司的故障风险、提高信息安全和技术。实施软件可靠性模型可以提高生产力,减少维护费用和故障成本。确保软件无故障是软件应用程序开发行业的一个重要问题。软件质量涵盖了一些关键因素,如软件可靠性、可维护性、可用性、在恶劣环境下的性能、稳健性以及对用户的数据保护。其中,软件可靠性是最重要的尺度;如果开发者不注意,它可能会影响公司的发展。软件可靠性关注所有类型的错误和故障,这些错误和故障可能影响任何类型的软件应用程序提供预期的解决方案的过程。对软件开发来说,确定软件的确切可靠性并不是一件容易的事。

对软件可靠性建立正确的模型并预测其可能的增长趋势,对于整个软件的可靠性至关重要[1] [2] [3]。软件可靠性增长模型 SRGM 已有众多模型被提出。在当前研究中,非齐次泊松过程(non-homogeneous Poisson process)类软件可靠性增长模型应用最为广泛;其中 G-O 模型[4]最为经典,其假设软件系统被检查出的累计故障数是泊松分布,并指出软件错误的内在数量是随机变量,对软件可靠性模型的开发作出了重要贡献。论文[5] [6] [7]中先后对 G-O 模型的假设作了不同的修改、完善,并提出了各种不同的模型。这些模型的应用需要基本的假设,如故障之间的时间独立,检测到的故障立即被纠正,以及故障纠正过程不引入新的故障。然而,其中一些假设似乎是不现实的[8]。此外,SRGM 的适用性已成为一个关键问题,因为没有单一模型可以在所有情况下普遍使用[9]。关于 SRGM 的可预测性,作为使用 SRGM 的主要目标,这些模型已经被证明能够对过去的软件故障数据进行建模和拟合,然而,它们并没有给出准确的预测。

因此一些研究人员试图将时间序列模型,特别是 ARIMA 模型应用于可靠性预测。主要结果是,时间序列模型可以解决 SRGM 的问题,而且它们有能力给出更准确的预测结果。此外,Junhong 等人[10]证明了软件可靠性预测中最常用的 Goel-Okumoto NHPP 模型可以转化为一阶自回归模型(AR(1))。这一结果说明,最重要的软件可靠性模型是 ARIMA 模型的一个特例,这为如何在软件可靠性预测中利用 ARIMA 模型的优势开辟了一个新的研究领域。胡和谢[11]讨论了不同类型的 ARIMA 模型,用于预测软件系统的故障。作者将 ARIMA 模型与 Duane 模型进行了比较。Duane 模型是基于非齐次泊松过程实现的,ARIMA 是一个线性模型,它是针对 Duane 模型提出的,作者对这些模型进行了分析以预测软件系统的数据故障。最后 ARIMA 模型被认为是更好的预测模型。Fan 和 Fan [12]提出了用时间序列评估工业软件系统的软件

可靠性分析和故障预测方案。研究发现，时间序列模型是一种可行的替代方法，就其对建筑设备可靠性的预测性能而言，对点故障和间隔故障的预测都能给出满意的结果。Amin 等人[13]提出了软件可靠性增长预测中的时间序列模型，作者认为在预测系统中使用不现实的假设在可靠性预测中没有意义，故提出了一种成熟的预测方法，它可以帮助软件可靠性工程师以更简单的方式构建正确的预测模型，从而提供相对于其他现有方法更准确的可靠性预测结果。Cao [14]提供了一个结合 ARIMA 模型与分形的线性和非线性建模的混合方案，并提出了一种研究软件失效机制的新思路。Choudhary [15]提出了使用机器学习技术或时间序列建模的非参数化软件可靠性增长模型。Pati 和 Shukla [16]提出将神经网络和求和自回归滑动平均模型的混合算法，这种整合有助于获取线性和非线性的故障模式。在这项工作中，线性和非线性部分被分别用来评估，以提高软件的可靠性。Chen 等人[17]提出了一种基于深度学习的方法，利用时间序列数据进行设备可靠性分析。Wang 等人[18]提出一种改进的时间序列方法应用于电力系统可靠性，并与传统的时间序列算法和蒙特卡洛方法进行比较。Kumaresan 等[19]提出了成熟的统计时间序列 (S)ARIMA 方法来开发一个预测模型，作者首先根据数据结构，对模型的适配能力进行评估。如果数据集的行为是线性的，则会选择 ARIMA 模型。如果是非线性的，那么(S)ARIMA 模型将被采用。在进行 (S)ARIMA 模型之前，它要找到系统中的季节性。Kamlesh 等[20]在季节性 ARIMA 模型基础上提出了一个基于故障分类的故障预测模型。综上，目前时间序列可靠性模型仅将软件看作一个整体，即使用软件的总故障数据进行建模，对于包含多组件的复杂软件，并未考虑组件之间的相关性以及对组件分别进行建模。

基于以上所述，本文的突破在于利用 VAR 模型建立了一个多变量约束的软件可靠性模型，以克服目前软件可靠性分析中单变量模型的缺陷，以及在多变量模型的情况下没有考虑成分间相关性的影响。所提出的基于向量自回归的软件可靠性预测模型被应用于两个真实的软件故障数据集，并与用于软件可靠性分析的 ARIMA 和 G-O 模型进行比较，在数据集 1 中，本文提出模型对于 G-O 模型和 ARIMA 模型拟合相对精度分别提高了 42.23% 和 30.88%；在数据集 2 中，拟合相对精度分别提高了 6.038% 和 1.299%。对于预测效果而言，在两个数据集中，对于 G-O 模型的相对精度分别提高了 37.893% 和 18.176%，对于 ARIMA 模型分别提高了 1.076% 和 12.418%。因此认为本文所提出的多元时间序列分析方法可以更完整、更全面地描述软件的可靠性，更准确地预测软件故障的发生。

## 2. 基于 VAR 模型的软件可靠性分析方法

### 2.1. VAR 模型预测结构

本文选用 Christopher Sims 在 1980 年[21]提出的向量自回归模型作为实证研究的方法，并利用 R 软件进行建模。

VAR 模型的数学表达式为：

$$z_t = \Phi_0 + \Phi_1 z_{t-1} + \dots + \Phi_p z_{t-p} + a_t \tag{1}$$

式(1)中： $\Phi_0$  是一个  $k$  维常数向量，并且对于  $i > 0$ ， $\Phi_i$  是  $k \times k$  维矩阵， $\Phi_p \neq 0$ ， $a_t$  是独立同分布(iid)随机向量序列，其均值为 0，协方差矩阵为正定矩阵。

可将 VAR 模型展开为如下形式：

$$\begin{pmatrix} z_{1t} \\ z_{2t} \\ \vdots \\ z_{kt} \end{pmatrix} = \begin{pmatrix} \phi_{10} \\ \phi_{20} \\ \vdots \\ \phi_{k0} \end{pmatrix} + \begin{pmatrix} \phi_{11}(1) & \dots & \phi_{1k}(1) \\ \vdots & \ddots & \vdots \\ \phi_{k1}(1) & \dots & \phi_{kk}(1) \end{pmatrix} \begin{pmatrix} z_{1t-1} \\ z_{2t-1} \\ \vdots \\ z_{kt-1} \end{pmatrix} + \dots + \begin{pmatrix} \phi_{11}(p) & \dots & \phi_{1k}(p) \\ \vdots & \ddots & \vdots \\ \phi_{k1}(p) & \dots & \phi_{kk}(p) \end{pmatrix} \begin{pmatrix} z_{1t-p} \\ z_{2t-p} \\ \vdots \\ z_{kt-p} \end{pmatrix} + \begin{pmatrix} a_{1t} \\ a_{2t} \\ \vdots \\ a_{kt} \end{pmatrix} \tag{2}$$

VAR 模型将所有变量纳入模型并对变量之间的关系进行更全面的分析。其一般流程为：1) 观察一段时间内的实验数据，检查并满足 VAR 模型的平稳性假设；2) 确定适当的 VAR 模型来描述这些数据；3) 使用估计方法拟合确定的模型；4) 检查估计模型的适当性；最后 5) 使用构建的预测模型来预测未来值。

## 2.2. 基于 VAR 模型的软件可靠性建模

本节将构造基于 VAR 模型的软件可靠性模型，将软件组件的故障数据代入(1)式得如下式(3)：

$$\begin{pmatrix} F_{1,t} \\ F_{2,t} \\ \vdots \\ F_{3,t} \end{pmatrix} = \Phi_0 + \Phi_1 \begin{pmatrix} F_{1,t-1} \\ F_{2,t-1} \\ \vdots \\ F_{3,t-1} \end{pmatrix} + \dots + \Phi_p \begin{pmatrix} F_{1,t-p} \\ F_{2,t-p} \\ \vdots \\ F_{3,t-p} \end{pmatrix} + a_t \quad (3)$$

其中  $F_i, i=1,2,\dots,n$  代表第  $i$  个组件。

## 2.3. 参数估计

### 2.3.1. 模型定阶

VAR 模型滞后期的选择对模型分析的准确性至关重要。较小的  $k$  值会导致误差项的严重自相关，而  $k$  值过大会导致自由度的减少，大大影响参数估计的有效性。在本文中，将综合考虑各准则来进行自由度的选择。3 个准则函数常用于决定 VAR 模型的阶数。VAR( $\varphi$ ) 模型的 3 个准则为：

$$AIC(\varphi) = \ln \left| \hat{\Sigma}_{a,\varphi} \right| + \frac{2}{T} \varphi k^2 \quad (4)$$

$$BIC(\varphi) = \ln \left| \hat{\Sigma}_{a,\varphi} \right| + \frac{\ln(T)}{T} \varphi k^2 \quad (5)$$

$$HQ(\varphi) = \ln \left| \hat{\Sigma}_{a,\varphi} \right| + \frac{2 \ln[\ln(T)]}{T} \varphi k^2 \quad (6)$$

其中  $T$  为样本数， $\hat{\Sigma}_{a,\varphi}$  为模型残差的协方差阵  $\Sigma_a$  的极大似然估计。

### 2.3.2. 系数矩阵估计

最小二乘法常被用来估计 VAR 模型的参数。即式(7)所示的估计值：

$$\hat{\beta} = (X'X)^{-1} (X'Z) = \left[ \sum_{t=p+1}^T x_t x_t' \right]^{-1} \sum_{t=p+1}^T x_t z_t' \quad (7)$$

其中  $x_t = (1, z'_{t-1}, \dots, z'_{t-p})'$  是一个  $(kp+1)$  维向量。

## 2.4. 模型评估准则

我们将本文提出的方法与 G-O 模型以及时间序列 ARIMA 模型进行对比评估。选取广泛应用于评估模型拟合优度的均方误差(MSE)，模型的 MSE 定义为：

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (8)$$

$y_t$  和  $\hat{y}_t$  分别为实际值与估计值。此外，我们使用  $RAI_{MSE}$  指标来衡量相对预测精度的提高，该指标定义为：

$$RAI_{MSE} - 1 = \frac{MSE_{ARIMA} - MSE_{VAR}}{MSE_{ARIMA}} \times 100 \quad (9)$$

$$RAI_{MSE-2} = \frac{MSE_{G-O} - MSE_{VAR}}{MSE_{G-O}} \times 100 \tag{10}$$

其中  $MSE_{VAR}$  和  $MSE_{ARIMA}$  以及  $MSE_{G-O}$  分别表示本文提出的方法、ARIMA 方法和 G-O 模型所产生的 MSE 值。

### 3. 案例分析

#### 3.1. 数据介绍

##### 3.1.1. 数据集 1

论文选取两组数据并展开实验研究，各组件故障数据均来源于用户缺陷跟踪系统。数据集 1 为 Kernel 软件的时间序列数据。Kernel 软件是操作系统的核心。本文选用数据集包括 input Devices、video、Bluetooth 以及 Network 四个组件在 2013 年 1 月~2022 年 8 月的故障数据，其中前 104 组数据用于训练模型，105~116 组的故障数据用于预测。input Devices、video、Bluetooth 以及 Network 四个组件分别用  $X_1, X_2, X_3, X_4$  表示具体数据见表 1:

**Table 1.** Component failure data for Kernel  
**表 1.** Kernel 的组件故障数据

时间/月	$X_1$	$X_2$	$X_3$	$X_4$	时间/月	$X_1$	$X_2$	$X_3$	$X_4$
1	1	0	1	0	59	4	2	3	4
2	2	2	0	0	60	2	3	2	2
3	0	2	0	0	61	9	3	2	3
4	0	2	0	0	62	2	3	6	5
5	2	1	0	0	63	2	3	6	3
6	0	0	0	0	64	0	6	6	3
7	0	0	0	0	65	2	5	5	5
8	1	1	1	0	66	5	6	3	2
9	1	0	0	0	67	4	9	1	2
10	2	2	0	0	68	7	5	1	1
11	3	9	3	1	69	7	6	1	0
12	4	5	2	0	70	8	4	2	1
13	1	4	0	0	71	7	3	1	3
14	0	3	4	0	72	6	6	2	1
15	1	0	1	0	73	6	3	5	2
16	1	3	3	0	74	3	4	0	1
17	1	2	1	0	75	3	7	4	3
18	2	1	1	0	76	4	7	4	3
19	0	3	8	0	77	2	2	3	1
20	6	5	0	0	78	8	4	1	1
21	2	1	0	0	79	4	6	1	2

## Continued

22	6	2	1	0	80	5	3	4	2
23	1	1	4	0	81	2	3	2	1
24	2	2	2	0	82	2	2	1	0
25	1	5	1	0	83	7	3	1	3
26	3	2	1	0	84	2	3	1	2
27	5	1	1	1	85	2	9	1	6
28	1	0	1	0	86	5	7	1	2
29	2	3	2	0	87	2	3	6	1
30	4	1	2	0	88	2	9	0	3
31	4	3	2	1	89	4	13	4	10
32	3	0	2	0	90	2	6	7	7
33	2	1	2	0	91	6	10	5	9
34	2	2	2	0	92	3	4	5	5
35	3	2	2	1	93	5	6	2	1
36	3	1	2	0	94	4	8	4	2
37	5	4	1	1	95	2	9	4	5
38	5	16	4	20	96	2	10	4	5
39	5	100	5	131	97	3	9	2	2
40	5	7	4	4	98	5	2	1	3
41	3	9	1	4	99	4	20	2	1
42	2	13	1	1	100	5	7	2	3
43	1	2	2	4	101	5	7	2	1
44	3	10	1	2	102	3	6	2	2
45	2	5	1	4	103	3	8	2	2
46	7	4	3	0	104	2	5	4	2
47	11	6	2	6	105	0	10	2	6
48	9	3	1	4	106	3	4	1	0
49	6	3	5	0	107	4	5	3	3
50	4	4	3	2	108	2	4	6	2
51	6	5	4	3	109	8	4	2	4
52	1	5	1	3	110	4	2	3	7
53	2	3	0	2	111	3	4	5	2
54	2	3	1	2	112	4	2	1	7
55	1	2	1	0	113	4	2	5	4
56	7	4	0	2	114	3	4	2	7
57	4	3	0	1	115	9	0	2	7
58	1	8	2	1	116	5	12	5	8

### 3.1.2. 数据集 2

数据集 2 为 *Akonadi* 软件 2009 年 1 月~2022 年 8 月的三个组件的故障数据, 将前 150 组数据划分为训练集; 剩余 14 组数据为测试集用于验证模型。*Akonadi* 是 *linux KDE* 中的一种以 *mysql* 进行数据库管理工作的 KDE 4 数据库接口。数据集 2 包含三个组件, 组件名称分别为 *general*, *IMAP* 以及 *server*, 在本文中用  $F_1, F_2$  以及  $F_3$  表示, 如表 2 所列:

**Table 2.** Component failure data for *Akonadi*  
**表 2.** *Akonadi* 的组件故障数据

时间/月	$F_1$	$F_2$	$F_3$	时间/月	$F_1$	$F_2$	$F_3$	时间/月	$F_1$	$F_2$	$F_3$
1	0	4	4	56	0	7	5	111	1	3	1
2	0	0	2	57	1	5	3	112	3	3	2
3	0	0	4	58	2	10	1	113	2	1	0
4	0	1	0	59	2	32	4	114	1	3	2
5	0	0	2	60	4	24	11	115	2	4	2
6	0	0	0	61	2	6	2	116	10	8	6
7	9	0	1	62	1	10	1	117	4	3	4
8	6	5	3	63	0	12	2	118	27	56	12
9	3	5	2	64	5	4	5	119	1	2	0
10	0	7	2	65	6	9	6	120	8	11	3
11	4	2	0	66	2	1	1	121	4	2	6
12	0	2	0	67	0	1	1	122	1	1	1
13	4	4	2	68	1	4	2	123	5	1	4
14	9	2	3	69	0	2	0	124	4	4	3
15	12	0	2	70	0	1	1	125	0	1	2
16	8	2	1	71	0	3	0	126	2	3	2
17	2	4	0	72	1	2	1	127	3	1	4
18	18	9	6	73	0	1	0	128	1	3	1
19	4	2	1	74	0	0	0	129	3	0	4
20	23	2	2	75	3	2	2	130	3	3	5
21	2	4	3	76	1	2	0	131	1	1	5
22	4	3	2	77	0	0	1	132	3	1	2
23	12	2	3	78	1	2	0	133	0	2	4
24	6	26	4	79	4	0	0	134	1	0	3
25	5	3	1	80	0	1	4	135	1	0	14
26	6	2	1	81	0	2	0	136	0	1	3
27	3	0	2	82	2	5	2	137	2	2	4
28	5	4	2	83	0	1	1	138	3	1	7
29	2	1	0	84	2	1	0	139	2	1	3

Continued

30	0	0	0	85	1	1	3	140	4	9	11
31	9	13	2	86	2	2	3	141	3	9	4
32	2	9	0	87	0	25	1	142	3	0	9
33	8	44	2	88	1	2	1	143	2	6	1
34	4	8	0	89	0	2	3	144	1	5	11
35	2	10	0	90	0	2	3	145	7	5	8
36	2	18	0	91	2	2	0	146	2	0	1
37	9	20	0	92	3	1	2	147	25	5	21
38	3	24	0	93	12	9	6	148	1	3	2
39	9	15	1	94	5	0	5	149	4	3	1
40	2	1	1	95	3	0	0	150	3	1	2
41	8	5	0	96	1	0	3	151	0	5	6
42	2	5	0	97	124	58	20	152	1	4	5
43	4	6	0	98	2	0	7	153	1	2	4
44	2	4	1	99	1	2	8	154	3	1	3
45	2	16	0	100	3	5	1	155	3	1	0
46	7	4	1	101	1	2	1	156	1	1	2
47	0	8	0	102	3	8	1	157	4	5	4
48	6	7	1	103	1	0	1	158	0	2	6
49	4	8	0	104	1	2	2	159	5	0	2
50	4	27	1	105	3	3	3	160	1	3	4
51	8	11	3	106	2	1	0	161	2	2	5
52	1	2	0	107	0	0	0	162	2	0	7
53	1	4	0	108	0	1	0	163	6	1	4
54	2	20	3	109	7	4	1	164	1	3	4
55	4	4	1	110	13	27	7				

## 3.2. 参数估计结果

### 3.2.1. 数据集 1 结果

在构建 VAR 模型之前,给定的时间序列数据应该满足平稳性的基本假设。本文采用 ADF (Augmented Dickey-Fuller test) 检验, 得出 p 值为 0.01, 在显著性水平为 0.05 的情况下认为数据是平稳的。根据表 3 所示结果, 参考 AIC 准则、BIC 准则及 HQ 准则等统计量, 最终选择的 VAR 模型最优滞后阶数为 1 阶。

因此对 Kernel 软件的四个组件 input Devices、video、Bluetooth 以及 Network 建立 VAR (1)模型, 通过参数估计得到拟合 VAR 模型如下:

$$\begin{pmatrix} X_{1,t} \\ X_{2,t} \\ X_{3,t} \\ X_{4,t} \end{pmatrix} = \begin{pmatrix} 0.353 & 0.059 & 0.142 & -0.036 \\ 0.400 & 0.066 & 0.655 & 0.049 \\ 0.164 & -0.023 & 0.239 & 0.033 \\ 0.547 & -0.089 & 0.853 & 0.181 \end{pmatrix} \begin{pmatrix} X_{1,t-1} \\ X_{2,t-1} \\ X_{3,t-1} \\ X_{4,t-1} \end{pmatrix} + \begin{pmatrix} 1.731 \\ 1.971 \\ 1.180 \\ -0.521 \end{pmatrix} \quad (11)$$



**Table 3.** Information criterion  
**表 3.** 信息准则

Lag	AIC	HQ	BIC
1	12.87748	13.18651	13.64028
2	12.94386	13.51042	14.34233
3	12.99026	13.81435	15.02441
4	13.27734	14.35896	15.94716
5	13.43538	14.77453	16.74087

### 3.2.2. 数据集 2 结果

经检验数据集 2 满足平稳性条件，根据表 4 所示，选择的 VAR 模型最优滞后阶数为 1 阶。

**Table 4.** Information criterion  
**表 4.** 信息准则

Lag	AIC	HQ	BIC
1	10.86253	10.95951	11.10126
2	10.86298	11.03269	11.28075
3	10.87380	11.11624	11.47061
4	10.92676	11.22194	11.68262
5	10.88484	11.27275	11.83975

因此对软件 *Akonadi* 的三个组件 *general*、*IMAP* 以及 *server* 建立 VAR (1)模型，通过参数估计得到拟合 VAR 模型如下：

$$\begin{pmatrix} F_{1,t} \\ F_{2,t} \\ F_{3,t} \end{pmatrix} = \begin{pmatrix} 0.014 & -0.044 & -0.082 \\ -0.076 & 0.217 & -0.364 \\ 0.01 & -0.046 & 0.197 \end{pmatrix} \begin{pmatrix} F_{1,t-1} \\ F_{2,t-1} \\ F_{3,t-1} \end{pmatrix} + \begin{pmatrix} 4.494 \\ 5.573 \\ 2.388 \end{pmatrix} \quad (12)$$

## 3.3. 模型性能对比分析

### 3.3.1. 训练集数据模型对比

通过 3.2 节建立的模型，使用第 3.1 节中提出的实际数据的训练集数据部分，我们得到了表 5 中模型的均方误差(MSE)和相对精度( $RAI_{MSE}$ )。在数据集 1 中，对比 G-O 模型，VAR 模型虽在组件  $X_1$  与  $X_3$  上的拟合效果略低，但 VAR 模型综合  $MSE(7.22)$  比 G-O 模型的综合  $MSE(68.61)$  小，这说明 VAR 模型的整体拟合误差小，且其相对精度提高了 42.23%。在数据集 2 中，VAR 模型与 G-O 模型的平均均方误差分别为 68.98 与 70.39，相对精度提高了 6.038%。

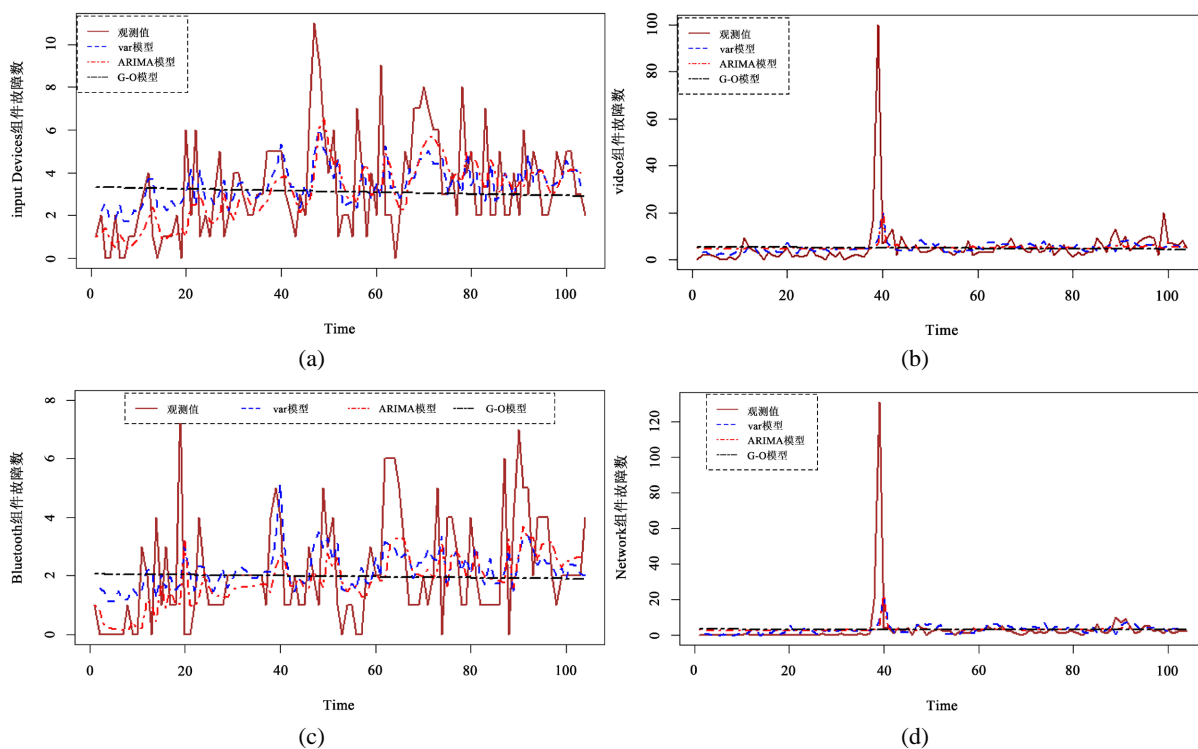
对比 ARIMA 模型，在数据集 1 中可知 VAR 模型在  $X_1$  以及  $X_3$  两个组件拟合的 MSE 值略大于 ARIMA 模型，但在数据集 1 中 VAR 模型相较于 ARIMA 模型相对精度提高了 30.88%，即在数据集 1 中两个模型各有优势；然而，从整体拟合效果来看，VAR 模型综合  $MSE(7.22)$  比 ARIMA 模型的综合  $MSE(60.24)$  小，这说明 VAR 模型的整体拟合误差小。而在数据集 2 中 VAR 模型除组件  $F_2$  的均方误差略低于 ARIMA 模型外，相对精度提高了 1.299%。总体来说，G-O 模型拟合效果最差，VAR 模型与 ARIMA 模型对各组

件故障数据的拟合各有优势，但从总体来说 VAR 模型更有优势。图 1 和图 2 分别给出了两个数据集中各个组件 VAR 模型、ARIMA 模型和 G-O 模型的月度故障数据的拟合图。

**Table 5.** MSE and  $RAI_{MSE}$  values of the three models in the training set

**表 5.** 训练集三种模型 MSE 与  $RAI_{MSE}$  值

数据集	组件	MSE			$RAI_{MAE_2}$		$RAI_{MAE_1}$	
		VAR	G-O	ARIMA	G-O	ARIMA		
1	$X_1$	5.76	5.48	4.37	-5.11	-31.81		
	$X_2$	11.22	99.75	87.80	88.75	87.22		
	$X_3$	3.61	3.29	2.86	-9.73	-26.22		
	$X_4$	8.29	165.92	145.93	95.004	94.32		
	Average	7.22	68.61	60.24	42.23	30.88		
2	$F_1$	115.39	115.63	116.64	0.208	1.072		
	$F_2$	81.58	83.73	80.46	2.568	-1.392		
	$F_3$	9.99	11.80	10.43	15.339	4.219		
	Average	68.98	70.39	69.18	6.038	1.299		



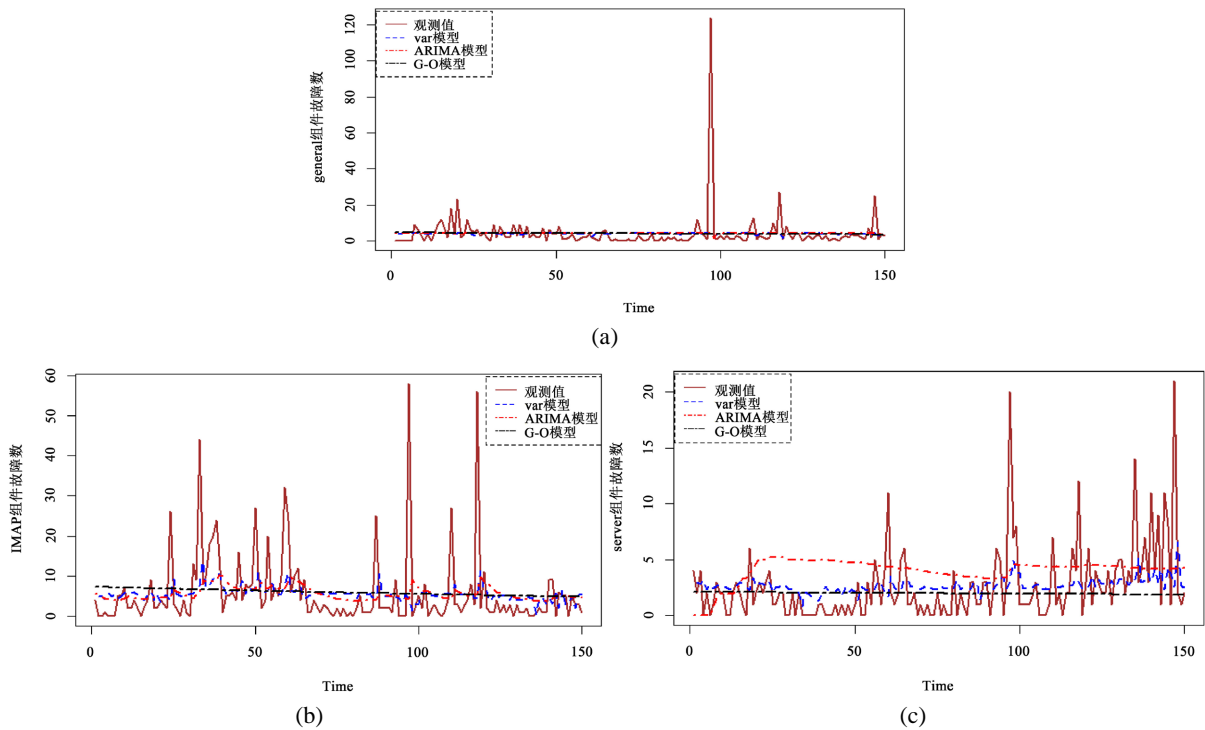
**Figure 1.** Fitting plot of the number of component failures for data set 1

**图 1.** 数据集 1 组件故障数拟合图

### 3.3.2. 测试集数据对比

对划分为测试集的月份使用上文建立的模型进行预测，并与测试集原始数据进行对比，得到两个数据集的 MSE 值，如表 6 所示。从表中可以看出，本文所提方法所给出的 MSE 值平均比 ARIMA 模型分

别低 1.076%与 12.418%，比 G-O 模型分别低 37.893%和 18.176%，这意味着本文方法提供了更准确的预测。图 3 和图 4 分别为两个数据集的 VAR 模型预测图。



**Figure 2.** Fitting plot of the number of component failures for data set 2  
**图 2.** 数据集 2 组件故障数拟合图

**Table 6.** MSE and  $RAI_{MSE}$  values of the three models in the test set  
**表 6.** 测试集三种模型 MSE 与  $RAI_{MSE}$  值

数据集	组件	MSE			$RAI_{MAE_2}$		$RAI_{MAE_1}$
		VAR	G-O	ARIMA	G-O	ARIMA	
1	$X_1$	3.168	7.078	4.529	55.242	30.051	
	$X_2$	3.497	10.564	9.11	66.897	61.614	
	$X_3$	4.884	4.340	2.368	-12.535	-106.25	
	$X_4$	5.664	9.760	6.893	41.967	18.889	
	Average	4.303	7.934	5.748	37.893	1.076	
2	$F_1$	7.54	8.03	8.58	6.102	12.12	
	$F_2$	11.79	14.02	15.76	15.91	25.19	
	$F_3$	5.250	7.78	5.248	32.519	-0.057	
	Average	8.19	9.94	9.86	18.176	12.418	

#### 4. 总结

软件的可靠性直接或间接取决于许多因素，并随着时间的推移而有很大的变化。现代软件可靠性的变化特征很难用传统的方式来全面描述，而且软件可靠性估计结果的差异性也比较大。基于对当前软件

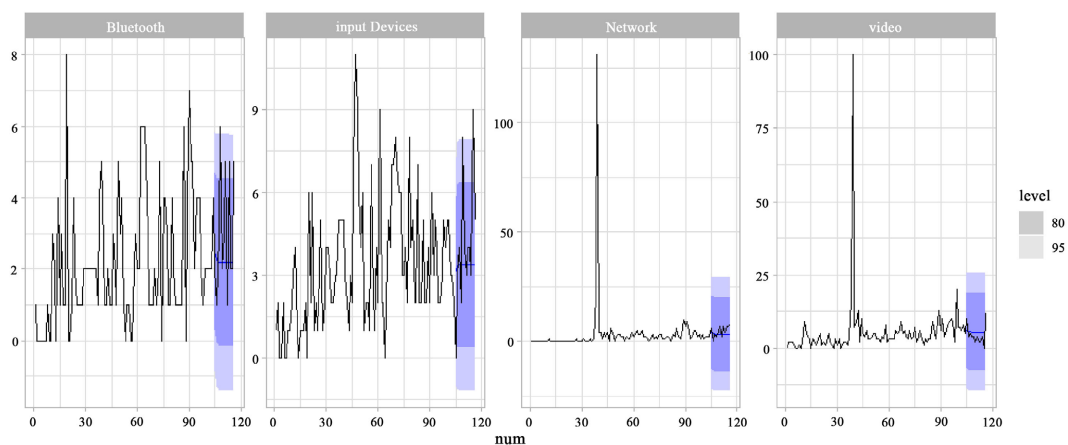


Figure 3. Prediction plots for dataset 1  
图 3. 数据集 1 的预测图

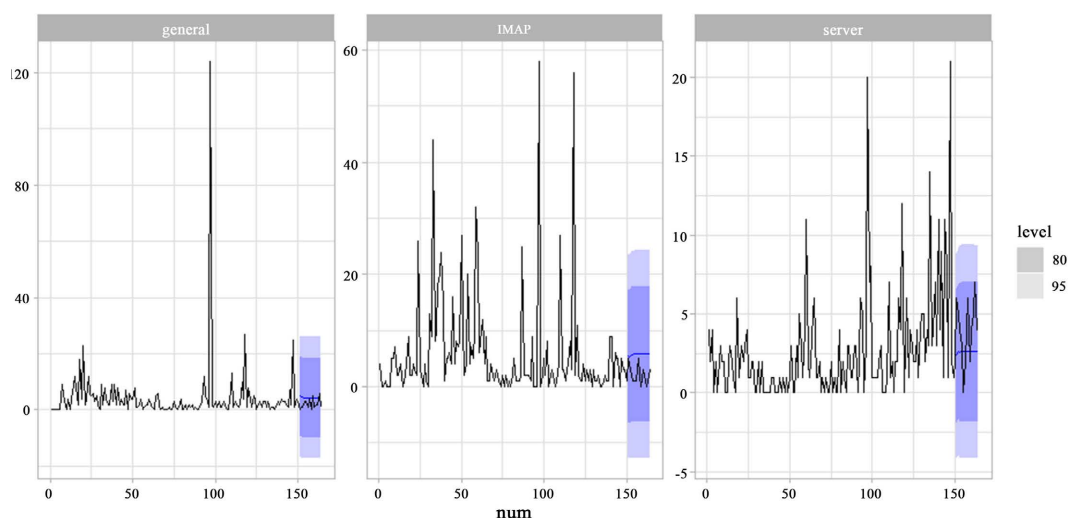


Figure 4. Prediction plots for dataset 2  
图 4. 数据集 2 的预测图

可靠性变化特征的分析, 采用向量自回归 VAR 模型来拟合各软件组件的故障数据, 建立预测模型。利用相同的数据, 在相同环境下与传统的 G-O 模型和 ARIMA 时间序列模型进行了比较; 结果表明, 这种方法提高了软件可靠性预测的准确性, 是一种优秀的软件可靠性预测方法。

## 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] Ahmad, N., Khan, M.G.M. and Rafi, L. (2010) A Study of Testing-Effort Dependent Inflection S- Shaped Software Reliability Growth Models with Imperfect Debugging. *International Journal of Quality & Reliability Management*, **27**, 89-110. <https://doi.org/10.1108/02656711011009335>
- [2] Okamura, H. and Dohi, T. (2014) A Novel Framework of Software Reliability Evaluation with Software Reliability Growth Models and Software Metrics. 2014 *IEEE 15th International Symposium on High-Assurance Systems Engineering*, Miami Beach, 9-11 January 2014, 97-104. <https://doi.org/10.1109/HASE.2014.22>

- [3] Okamura, H., Etani, Y. and Dohi, T. (2011) Quantifying the Effectiveness of Testing Efforts on Software Fault Detection with a Logit Software Reliability Growth Model. 2011 *Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement*, Nara, 3-4 November 2011, 62-68. <https://doi.org/10.1109/IWSM-MENSURA.2011.26>
- [4] Goel, A.L. and Okumoto, K. (1979) Time-Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures. *IEEE Transactions on Reliability*, **R-28**, 206-211. <https://doi.org/10.1109/TR.1979.5220566>
- [5] Huang, C.-Y., Lyu, M.R. and Kuo, S.-Y. (2003) A Unified Scheme of Some Non-Homogenous Poisson Process Models for Software Reliability Estimation. *IEEE Transactions on Software Engineering*, **29**, 261-269. <https://doi.org/10.1109/TSE.2003.1183936>
- [6] Yamada, S., Ohba, M. and Osaki, S. (1983) S-Shaped Reliability Growth Modeling for Software Error Detection. *IEEE Transactions on Reliability*, **R-32**, 475-478. <https://doi.org/10.1109/TR.1983.5221735>
- [7] Zhao, J., Liu H-W, Cui, G. and Yang, X.-Z. (2006) Software Reliability Growth Model with Change-Point and Environmental Function. *Journal of Systems and Software*, **79**, 1578-1587. <https://doi.org/10.1016/j.jss.2006.02.030>
- [8] Goel, A.L. (1985) Software Reliability Models: Assumptions, Limitations, and Applicability. *IEEE Transactions on Reliability*, **SE-11**, 1411-1423. <https://doi.org/10.1109/TSE.1985.232177>
- [9] Sharma, K., Garg, R., Nagpal, C.K. and Garg, R.K. (2010) Selection of Optimal Software Reliability Growth Models Using a Distance Based Approach. *IEEE Transactions on Reliability*, **59**, 266-276. <https://doi.org/10.1109/TR.2010.2048657>
- [10] Guo, J.H., Liu, H.W. and Yang, X.Z. (2005) An Autoregressive Time Series Software Reliability Growth Model with Independent Increment. *Mathematical Methods and Computational Techniques in Electrical Engineering*, 362-366.
- [11] Ho, S.L. and Xie, M. (1998) The Use of Arima Models for Reliability Forecasting and Analysis. *Computers and Industrial Engineering*, **35**, 213-216. [https://doi.org/10.1016/S0360-8352\(98\)00066-7](https://doi.org/10.1016/S0360-8352(98)00066-7)
- [12] Fan, Q. and Fan, H. (2015) Reliability Analysis and Failure Prediction of Construction Equipment with Time Series Models. *Journal of Advanced Management Science*, **3**, 203-210. <https://doi.org/10.12720/joams.3.3.203-210>
- [13] Amin, A., Grunske, L. and Colman, A. (2013) An Approach to Software Reliability Prediction Based on Time Series Modeling. *Journal of Systems and Software*, **86**, 1923-1932. <https://doi.org/10.1016/j.jss.2013.03.045>
- [14] Cao, Y. and Zhu, Q. (2010) The Software Reliability Model Using Hybrid Model of Fractals and Arima. *IEICE Transactions on Information and Systems*, **E93-D**, 3116-3119. <https://doi.org/10.1587/transinf.E93.D.3116>
- [15] Choudhary, A., Baghel, A.S. and Sangwan, O.P. (2016) Software Reliability Prediction Modeling: A Comparison of Parametric and Non-Parametric Modeling. 2016 *6th International Conference - Cloud System and Big Data Engineering (Confluence)*, Noida, 14-15 January 2016, 649-653 <https://doi.org/10.1109/CONFLUENCE.2016.7508198>
- [16] Pati, J. and Shukla, K.K. (2015) A Hybrid Technique for Software Reliability Prediction. *Proceedings of the 8th India Software Engineering Conference*, Bangalore, 18-20 February 2015, 139-146. <https://doi.org/10.1145/2723742.2723756>
- [17] Chen, B., Liu, Y., Zhang, C. and Wang, Z. (2020) Time Series Data for Equipment Reliability Analysis with Deep Learning. *IEEE Access*, **8**, 105484-105493. <https://doi.org/10.1109/ACCESS.2020.3000006>
- [18] Wang, Z., Hu, Z. and Yang, X. (2020) Research on Power System Reliability Evaluation Based on Improved Time Series Algorithm. 2020 *Chinese Automation Congress (CAC 2020)*, Shanghai, 6-8 November 2020, 3669-3673. <https://doi.org/10.1109/CAC51589.2020.9326621>
- [19] Kumaresan, K. and Ganeshkumar, P. (2020) Software Reliability Prediction Model with Realistic Assumption Using Time Series (S)Arima Model. *Journal of Ambient Intelligence and Humanized Computing*, **11**, 5561-5568. <https://doi.org/10.1007/s12652-020-01912-4>
- [20] Raghuvanshi K K, Agarwal, A., Jain, K. and Singh, V.B. (2021) A Generalized Prediction Model for Improving Software Reliability Using Time-Series Modelling. *International Journal of System Assurance Engineering and Management*, **13**, 1309-1320. <https://doi.org/10.1007/s13198-021-01449-5>
- [21] Sims, C.A. (1980) Macroeconomics and Reality. *Econometrica*, **48**, 1-48. <https://doi.org/10.2307/1912017>