Scientific Research Publishing

American Journal of Operations Research, 2022, 12, 1-10
https://www.scirp.org/journal/ajor
ISSN Online: 2160-8849
ISSN Print: 2160-8830

A Software Reliability Model for OSS Including Various Fault Data Based on Proportional Hazard-Rate Model

Taku Yanagisawa[1], Yoshinobu Tamura[2], Adarsh Anand[3], Shigeru Yamada[4]

[1]Tokyo City University, Tokyo, Japan
[2]Yamaguchi University, Yamaguchi, Japan
[3]University of Delhi, Delhi, India
[4]Tottori University, Tottori, Japan
Email: g2081444@tcu.ac.jp, tamuray@yamaguchi-u.ac.jp, adarsh.anand86@gmail.com, yamada@tottori-u.ac.jp

How to cite this paper: Yanagisawa, T., Tamura, Y., Anand, A. and Yamada, S. (2022) A Software Reliability Model for OSS Including Various Fault Based on Proportional Hazard-Rate Model. *American Journal of Operations Research*, **12**, 1-10.
https://doi.org/10.4236/ajor.2022.121001

Received: November 22, 2021
Accepted: January 2, 2022
Published: January 5, 2022

Abstract

The software reliability model is the stochastic model to measure the software reliability quantitatively. A Hazard-Rate Model is the well-known one as the typical software reliability model. We propose Hazard-Rate Models Considering Fault Severity Levels (CFSL) for Open Source Software (OSS). The purpose of this research is to make the Hazard-Rate Model considering CFSL adapt to baseline hazard function and 2 kinds of faults data in Bug Tracking System (BTS), *i.e.*, we use the covariate vectors in Cox proportional Hazard-Rate Model. Also, we show the numerical examples by evaluating the performance of our proposed model. As the result, we compare the performance of our model with the Hazard-Rate Model CFSL.

Keywords

Open Source Software, Fault Data, Software Reliability, Cox Proportional Hazard-Rate Model

1. Introduction

Open Source Software (OSS) is used by many organizations in various situations because of its low cost, standardization, and quick delivery. However, the quality of OSS is not ensured, because OSS is developed by many volunteers around the world in a unique development style. Then, the development style has no organized testing phase. The faults latent in OSS are usually fixed by using the database of Bug Tracking System (BTS). There is various information related to faults in BTS. The reliability assessment of OSS is necessary and important for the demand in the future and the current problem of OSS.

The software reliability model is a mathematical model to measure software reliability in statistical and stochastic approaches. As of today, many various models not only for proprietary software but also for OSS have been proposed by a lot of researchers [1]-[6]. The Hazard-Rate model is well known as the typical software reliability model [7] [8] [9] [10]. We proposed a Hazard-Rate Model Considering Fault Severity Levels (CFSL) for OSS in the past [11]. Mostly, a lot of Hazard-Rate Models measure the software reliability with only the data of the time of occurrence of software failures in the testing or operation phase. However, we can get various information related to faults of software aside from the data of the time of occurrence of software failures. As for previous research, the Hazard-Rate Model includes the data of the failure identification work and execution time in CPU, which are called environment data in the paper. Then, the related models have been proposed in the past by using Cox Proportional Hazard-Rate Model (Cox PHM) [12] [13]. Specifically, these models have been made the traditional Hazard-Rate Model adapt to baseline hazard function and the environment data to the covariate vectors in Cox PHM. On the other hand, the Hazard-Rate Model for OSS based on PHM with various faults data in BTS has not been proposed until today.

The purpose of our research is to propose the Hazard-Rate Model including various faults data in BTS of OSS. Specifically, we make the Hazard-Rate Model with CFSL adapt to the baseline hazard function in Cox PHM, and 2 kinds of faults data in BTS to the covariate vectors in Cox PHM. Moreover, we show several numerical examples based on the proposed model to evaluate the performance of the model.

2. Bug Tracking System

BTS is the database. This is that OSS users can report the information about faults in OSS. There is various information in BTS, e.g., the recorded time of fault, the time of fault to be fixed, the nickname of fault assignee, and so on. We show the list of fault data in BTS in **Table 1**.

3. Hazard-Rate Model

Firstly, we show the stochastic quantities related to the number of software faults and the time of occurrence of software failures in testing phase or operating phase as shown in **Figure 1**.

The distribution function of $X_k\left(k=1,2,\cdots\right)$ representing the time-interval between successive detected faults of $\left(k-1\right)^{\text{st}}$ and k^{th} is defined as:

$$F_k\left(x\right)\equiv \Pr\left\{X_k\le x\right\}\quad\left(x\ge 0\right)\tag{1}$$

where: Pr{A} represents the occurrence probability of event A. Therefore, the following derived function means the probability density function of X_k:

$$f_k\left(x\right)\equiv\frac{\mathrm{d}F_k\left(x\right)}{\mathrm{d}x}\tag{2}$$

Table 1. The list of kind of fault data in BTS.

The kind of fault data	Contents
Opened	The date and time recorded on the bug tracking system.
Changed	The modified date and time.
Product	The name of product included in OSS.
Component	The name of component included in OSS.
Version	The version number of OSS.
Reporter	The nickname of fault reporter.
Assignee	The nickname of fault assignee.
Severity	The level of fault.
Status	The fixing status of fault.
Resolution	The status of resolution of fault.
Hardware	The name of hardware under fault occurrence.
OS	The name of operating system under fault occurrence.
Summary	The brief contents of fault.

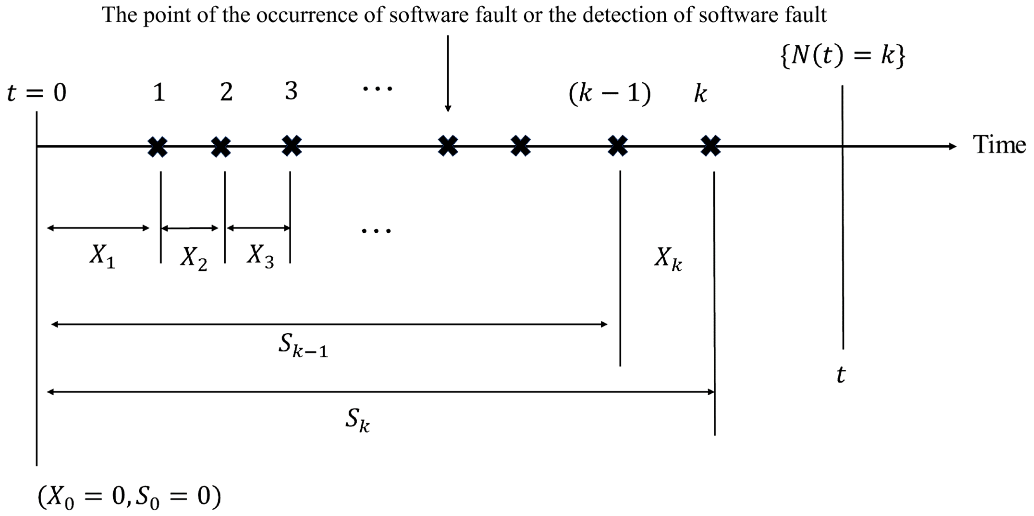

Figure 1. The variables of the software fault detection event and the software fault occurrence one.

Also, the software reliability can be defined as the probability that a software failure does not occur during the time-interval $(0, x]$. The software reliability is given by:

$$R_k(x) \equiv \Pr\{X_k > x\} = 1 - F_k(x) \tag{3}$$

From Equations (1)-(3), the hazard-rate is given by the following equation:

$$z_k(x) \equiv \frac{f_k(x)}{1 - F_k(x)} = \frac{f_k(x)}{R_k(x)} \tag{4}$$

where: the Hazard-Rate means the software failure rate when the software failure does not occur during the time-interval $(0, x]$. A Hazard-Rate Model is a software reliability model representing the software failure-occurrence phenomenon by the Hazard-Rate.

Moreover, we discuss three Hazard-Rate Models as follows.

3.1. Jelinski-Moranda Model

Jelinski-Moranda (J-M) model is one of the Hazard-Rate Models. J-M model has the following assumptions:

1) The software failure rate during a failure interval is constant and is proportional to the number of faults remaining in the software;

2) The number of remaining faults in the software decreases by one each time a software failure occurs;

3) Any fault that remains in the software has the same probability of causing a software failure at any time.

From the above assumptions, the software Hazard-Rate in Equation (4) at k^{th} can be derived as:

$$z_k(x) = \phi\left[N - (k-1)\right] \quad \left(N > 0, \phi > 0; k = 1, 2, \cdots, N\right) \tag{5}$$

where: each parameter is defined as follows:

N: the number of latent software faults before the testing;

ϕ: the Hazard-Rate per inherent fault.

3.2. Moranda Model

Moranda model has the following assumptions:

The software failure rate per software fault is constant and is decreasing geometrically as a fault is discovered.

From the above assumptions, the software Hazard-Rate in Equation (4) at k^{th} can be derived as:

$$z_k(x) = D \cdot c^{k-1} \quad \left(D > 0, 0 < c < 1; k = 1, 2, \cdots\right) \tag{6}$$

where each parameter is defined as follows:

D: the initial Hazard-Rate for the software failure;

c: the decrease coefficient for Hazard-Rate.

3.3. Xie Model

Xie model has the following assumptions:

The software failure rate per software fault is constant and is decreasing exponentially with the number of faults remaining in the software.

From the above assumptions, the software Hazard-Rate in Equation (4) at k^{th} can be derived as:

$$z_k(x) = \lambda_0 (N - k + 1)^{\alpha} \quad \left(N > 0, \lambda_0 > 0, \alpha \ge 1; k = 1, 2, \cdots, N\right) \tag{7}$$

where each parameter is defined as follows:

N: the number of latent software faults before the testing;

λ_0: the Hazard-Rate per inherent fault;

α : the constant parameter.

3.4. Mean Time between Failures (MTBF)

Three Hazard-Rate Models above have the following assumption:

Any fault that remains in the software have the same probability of causing s software failure at any time.

From this assumption, three Hazard-Rate Models are called exponential Hazard-Rate Model. MTBF by three Hazard-Rate Models can be derived as:

$$\mathrm{E}\left[X_k\right] = \int_0^\infty x f_k\left(x\right)\mathrm{d}x = \int_0^\infty R_k\left(x\right)\mathrm{d}x \equiv \frac{1}{z_k\left(x\right)} \tag{8}$$

4. Hazard-Rate Model Considering Fault Severity Levels (CFSL)

Hazard-Rate Model CFSL is the Hazard-Rate Model for OSS considering the fault severity levels in BTS. This model represents the Hazard-Rate for OSS itself by representing the Hazard-Rate for the normal fault and for the others one respectively. In this section, we discuss the Hazard-Rate Model CFSL.

We assume that the fault data is divided into the following types in terms of the fault severity levels in BTS:

A1: the normal fault;

A2: the others fault.

In the assumption above, A1 is the fault detected as a normal one, A2 is the fault detected as the other one. Also, OSS manager cannot differentiate between assumptions A1 and A2 in terms of the software faults. The time interval between successive faults of $\left(k-1\right)^{\text{st}}$ and k^{th} is represented as the random variable $X_k\left(k=1,2,\cdots\right)$, Therefore, the Hazard-Rate function $z_k\left(x\right)$ for X_k is defined as follows:

$$z_k\left(x\right) = p\cdot z_k^1\left(x\right) + \left(1-p\right)\cdot z_k^2\left(x\right) \quad \left(k=1,2,\cdots;0\le p\le 1\right) \tag{9}$$

$$z_k^1\left(x\right) = D_1\cdot c_1^{k-1} \quad \left(k=1,2,\cdots;D_1\ge 0,0<c_1<1\right) \tag{10}$$

$$z_k^2\left(x\right) = D_2\cdot c_2^{k-1} \quad \left(k=1,2,\cdots;D_2\ge 0,0<c_2<1\right) \tag{11}$$

where each parameter is defined as follows:

$z_k^1\left(x\right)$: the Hazard-Rate for assumption A1;

D_1 : the initial Hazard-Rate for the first software failure of A1;

c_1 : the decrease coefficient for Hazard-Rate for assumption A1;

$z_k^2\left(x\right)$: the Hazard-Rate for assumption A2;

D_2 : the initial Hazard-Rate for the first software failure of A2;

c_2 : the decrease coefficient for Hazard-Rate for assumption A2;

p: the weight parameter for $z_k^1\left(x\right)$.

Equation (10) represents the Hazard-Rate for a software failure-occurrence phenomenon for the normal fault. On the other hand, Equation (11) represents the Hazard-Rate for a software failure-occurrence for the other one.

5. Cox Proportional Hazard-Rate Model

Cox PHM is the model representing Hazard-Rate by using baseline hazard function, which is subject for a variable of time, and covariate vector. In this section, we discuss about Cox PHM.

It is assumed that two kinds of vectors are defined as follows:

$$\boldsymbol{\alpha}_k = \left(\alpha_{k1}, \alpha_{k2}, \cdots, \alpha_{kj}, \cdots, \alpha_{kq}\right) \quad (k = 1, 2, \cdots), \tag{12}$$

$$\boldsymbol{\beta} = \left(\beta_1, \beta_2, \cdots, \beta_j, \cdots, \beta_q\right), \tag{13}$$

where each vector is defined as follows:

$\boldsymbol{\alpha}_k$: the covariate vector including q kinds of data $\alpha_{kj}\left(j = 1, \cdots, q\right)$ for X_k;

$\boldsymbol{\beta}$: the coefficient vector for $\boldsymbol{\alpha}_k$.

Therefore, Cox PHM is defined as follows by using two vectors above:

$$h_k\left(x, \boldsymbol{\alpha}\right) = h_0\left(x_k\right)\exp\left(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\beta}\right) = h_0\left(x_k\right)\exp\left(\alpha_{k1}\beta_1 + \cdots + \alpha_{kq}\beta_q\right) \tag{14}$$

where: $h_0\left(x_k\right)$ in Equation (14) is called baseline hazard function and is subject for a variable of x_k.

6. Proposed Model

As a proposed model, we apply the Hazard-Rate Model CFSL to the baseline hazard function in the Cox PHM. Moreover, we use the assignee data in BTS and Mean Time Between Correction (MTBC) into the covariate vector. Then, our proposed model is derived as follows:

$$\begin{aligned} h_k\left(x, \boldsymbol{\alpha}\right) &= z_k\left(x\right)\exp\left(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\beta}\right) \\ &= \left\{p \cdot z_k^1\left(x\right) + \left(1 - p\right) \cdot z_k^2\left(x\right)\right\}\exp\left(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\beta}\right) \quad \left(k = 1, 2, \cdots; 0 \le p \le 1\right) \end{aligned} \tag{15}$$

where each parameter is defined as follows:

$z_k^1\left(x\right)$: the Hazard-Rate for assumption A1;

$z_k^2\left(x\right)$: the Hazard-Rate for assumption A2;

p: the weight parameter for $z_k^1\left(x\right)$;

$\boldsymbol{\alpha}_k$: the data of assignee and MTBC in OSS;

$\boldsymbol{\beta}$: the coefficient parameter for $\boldsymbol{\alpha}_k$.

In this paper, we apply the exponential Hazard-Rate Model to the baseline hazard function. Thus, the proposed model can be regarded as a parametric model. Moreover, the distribution function and the density function of X_k are derived as a Equation (16), (17) respectively.

$$F_k\left(x\right) = 1 - \exp\left(-\int_0^x z_k\left(x\right)\exp\left(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\beta}\right)\mathrm{d}x\right) \tag{16}$$

$$f_k\left(x\right) = z_k\left(x\right)\exp\left(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\beta}\right)\exp\left(-\int_0^x z_k\left(x\right)\exp\left(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\beta}\right)\mathrm{d}x\right) \tag{17}$$

For this reason, the parameters in the proposed model can be estimated by MLE (Maximum Likelihood Estimation).

7. Numerical Example

We use of fault big data in Apache HTTP server to estimate MTBF as the evaluation of the performance of our proposed model compared to Hazard-Rate Model CFSL [14]. The data of assignee is converted in numerical one in the form of frequency of occurrence. Specifically, our proposed model is divided into three cases as follows:

PHM1: the data of assignee is only included in $\boldsymbol{\alpha}_k$;

PHM2: MTBC is only included in $\boldsymbol{\alpha}_k$;

PHM3: the data of assignee and MTBC are included in $\boldsymbol{\alpha}_k$.

The parameters in the proposed models are estimated by MLE (Maximum Likelihood Estimation). The estimated value of parameters in three models is shown in Table 2.

In Table 2, $w_1 = pD_1$ and $w_2 = (1-p)D_2$ are assumed for the simplification technique. The vector α_{k1} includes the data of assignee and α_{k2} includes MTBC. Thus, $\widehat{\beta_1}$ and $\widehat{\beta_2}$ are the coefficient parameter for α_{k1} and α_{k2}, respectively. In this paper, we assume that the future data of assignee and MTBC are possible to be expected or already detected. The value of correlation coefficient between α_{k1} and α_{k2} is $r = 0.29$. Therefore, there is not multicollinearity in PHM3.

As a criterion to measure the goodness-of-fit of our proposed model, we use AIC (Akaike's Information based on the maximum likelihood estimation of model parameters Criterion).

Figures 2-5 show the estimated MTBF for each model and Table 3 shows the

Table 2. The estimated value of each parameter in the model.

	Value of Parameter					
Models	$\widehat{w_1}$	$\widehat{w_2}$	$\widehat{c_1}$	$\widehat{c_2}$	$\widehat{\beta_1}$	$\widehat{\beta_2}$
PHM1	2.15302	1.94836	0.99944	0.99994	−0.00088	-
PHM2	2.11500	2.20817	0.99993	0.99933	-	−8.18298e−06
PHM3	2.65362	1.89733	0.99942	0.99995	0.00898	−5.52184e−05

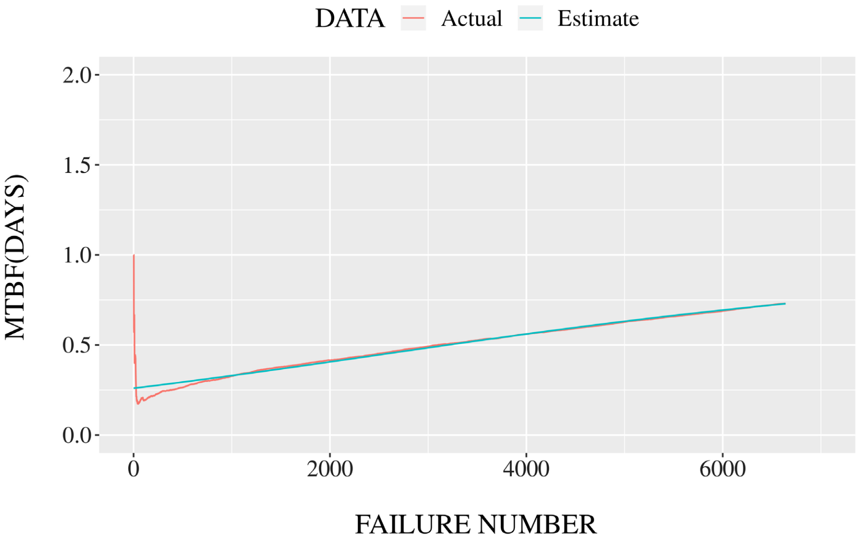

Figure 2. The estimated MTBF by using Hazard-Rate Model CFSL.

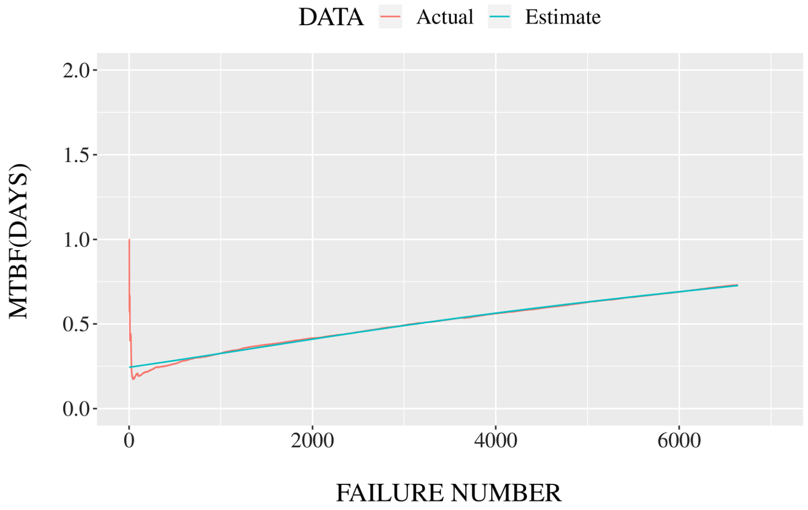

Figure 3. The estimated MTBF by using PHM1.

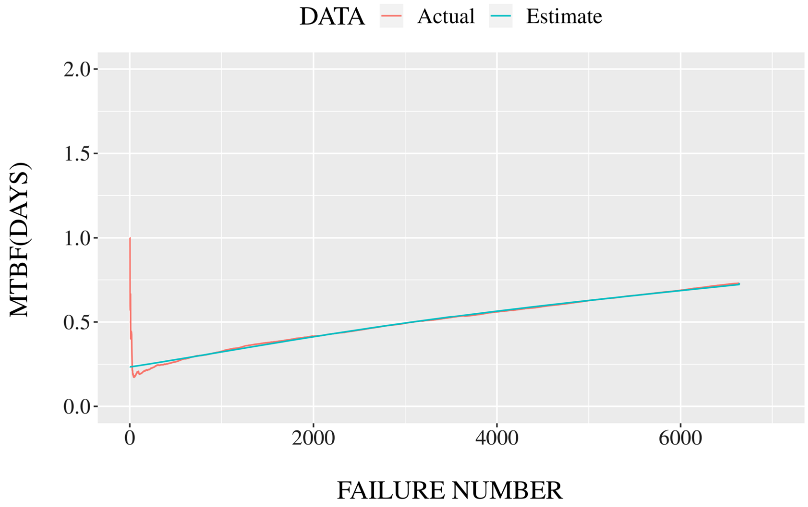

Figure 4. The estimated MTBF by using PHM2.

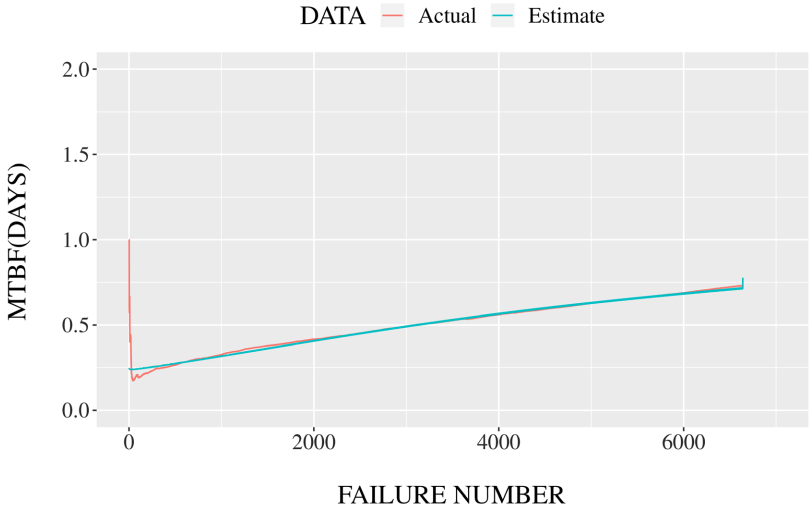

Figure 5. The estimated MTBF by using PHM3.

Table 3. The values of AIC for each model.

Model	AIC
Hazard-Rate Model CFSL	13,326.5
PHM1	13,322.8
PHM2	13,320.6
PHM3	13,322.2

value of AIC for each model. From Figures 2-5, PHM estimates MTBF shorter than Hazard-Rate Model CFSL at the initial faults slightly. In terms of AIC, we find that PHM fits better than the Hazard-Rate Model CFSL from Table 3. In other words, PHM is possible to predict the MTBF of OSS more correctly.

8. Conclusions

In this paper, we have proposed the Hazard-Rate Models for OSS including various fault data based on Cox PHM. Specifically, we have made the Hazard-Rate Models considering CFSL adapt to the baseline hazard function. Besides, we have applied the data of assignee and MTBC into the covariate vectors in Cox PHM. Also, we have shown numerical examples to evaluate the performance of our model. As the result, we have shown that the proposed model predicts MTBF, and fits better than the Hazard-Rate model considering CFSL in terms of AIC.

OSS is popular and in demand for a lot of organizations in various situations. However, OSS is developed by many volunteers in the world without an explicit testing phase. Therefore, the reliability of OSS is not ensured. For this reason, it is necessary to measure software reliability quantitatively. There are various fault data in the BTS of OSS. Then, the data sets are useful to find the characteristics of OSS. Moreover, we can assess software reliability accurately by using not only the data of the time of occurrence of software failures in the testing or operation phase but also the other various fault data in BTS.

In BTS, there are many kinds of fault data aside from the one we used in this paper. Therefore, we will discuss the proposal of other software reliability models with other kinds of fault data in BTS as future research. Also, we would like to suggest new measurements for OSS reliability including the characteristics of OSS.

Acknowledgements

This work was supported in part by the JSPS KAKENHI Grant No. 20K11799 in Japan.

Conflicts of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

References

[1] Tamura, Y. and Yamada, S. (2020) Large Scale Fault Data Analysis and OSS Relia-

bility Assessment Based on Quantification Method of the First Type, *Machine Learning and Knowledge Extraction*, **2**, 436-452. https://doi.org/10.3390/make2040024

[2] Yamada, S. and Sera, K. (1999) Imperfect Debugging Models with Two Kinds of Software Hazard Rate and Their Bayesian Formulation. *IEICE Transactions on Fundamentals*, **J82-A**, 1577-1584. (in Japanese)

[3] Tamura, Y. and Yamada, S. (2018) AI Approach to Fault Big Data Analysis and Reliability Assessment for Open-Source Software. In: Anand, A. and Ram, M., Eds., *System Reliability Management: Solutions and Technologies, Advanced Research in Reliability and System Assurance Engineering*. CRC Press Taylor & Francis Group, Boca Raton, 1-17. https://doi.org/10.1201/9781351117661-1

[4] Tamura, Y. and Yamada, S. (2013) Reliability Assessment Based on Hazard Rate Model for an Embedded OSS Porting Phase. *Software: Testing, Verification and Reliability*, **23**, 77-88. https://doi.org/10.1002/stvr.455

[5] Tamura, Y. and Yamada, S. (2010) Software Reliability Analysis with Optimal Release Problems Based on Hazard Rate Model for an Embedded OSS. 2010 *IEEE International Conference on Systems, Man and Cybernetics*, 720-726. https://doi.org/10.1109/ICSMC.2010.5641839

[6] Barack, O. and Huang, L. (2020) Assessment and Prediction of Software Reliability in Mobile Applications. *Journal of Software Engineering and Applications*, **13**, 179-190. https://doi.org/10.4236/jsea.2020.139012

[7] Jelinski, Z. and Moranda, P.B. (1972) Software Reliability Research. In: Freiberger, W. Ed., *Statistical Computer Performance Evaluation*, Academic Press, New York, 465-484. https://doi.org/10.1016/B978-0-12-266950-7.50028-1

[8] Monrada, P.B. (1979) Event-Altered Rate Models for General Reliability Analysis. *IEEE Transactions on Reliability*, **R-28**, 376-381. https://doi.org/10.1109/TR.1979.5220648

[9] Xie, M. (1989) On a Generalization of J-M Model. *Proceedings of Reliability*, **89**, 5.

[10] Schick, G.J. and Wolverton, R.W. (1978) An Analysis of Competing Software Reliability Models. *IEEE Transactions on Software Engineering*, **SE-4**, 104-120. https://doi.org/10.1109/TSE.1978.231481

[11] Yanagisawa, T., Tamura, Y., Anand, A. and Yamada, S. (2021) Comparison of Hazard-Rates Considering Fault Severity Levels and Imperfect Debugging for OSS. *Journal of Software Engineering and Applications*, **14**, 591-606. https://doi.org/10.4236/jsea.2021.1411035

[12] Nishio, Y., Dohi, T. and Osaki, S. (2002) A Reliability Assessment of Software Product Based on Proportional Hazards Models. *IEICE Transactions on Fundamentals*, **J85-A**, 84-94.

[13] Cox, D.R. (1972) Regression Models and Life Tables. *Journal of the Royal Statistical Society*, **B-34**, 187-220.

[14] The Apache Software Foundation (2021) The Apache HTTP Server Project. http://httpd.apache.org/