Scientific Research Publishing

American Journal of Operations Research, 2020, 10, 269-283
https://www.scirp.org/journal/ajor
ISSN Online: 2160-8849
ISSN Print: 2160-8830

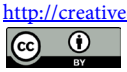
OSS Project Assessment Based on Discriminant Analysis and Jump Diffusion Process Model for Fault Big Data

Yoshinobu Tamura[1], Hayato Watanabe[1], Shigeru Yamada[2]

[1]Tokyo City University, Tokyo, Japan
[2]Tottori University, Tottori, Japan
Email: tamuray@tcu.ac.jp, yamada@tottori.ac.jp

How to cite this paper: Tamura, Y., Watanabe, H. and Yamada, S. (2020) OSS Project Assessment Based on Discriminant Analysis and Jump Diffusion Process Model for Fault Big Data. *American Journal of Operations Research*, **10**, 269-283.
https://doi.org/10.4236/ajor.2020.106015

Received: August 31, 2020
Accepted: November 6, 2020
Published: November 9, 2020



Abstract

The bug tracking system is well known as the project support tool of open source software. There are many categorical data sets recorded on the bug tracking system. In the past, many reliability assessment methods have been proposed in the research area of software reliability. Also, there are several software project analyses based on the software effort data such as the earned value management. In particular, the software reliability growth models can apply to the system testing phase of software development. On the other hand, the software effort analysis can apply to all development phase, because the fault data is only recorded on the testing phase. We focus on the big fault data and effort data of open source software. Then, it is difficult to assess by using the typical statistical assessment method, because the data recorded on the bug tracking system is large scale. Also, we discuss the jump diffusion process model based on the estimation method of jump parameters by using the discriminant analysis. Moreover, we analyze actual big fault data to show numerical examples of software effort assessment considering many categorical data set.

Keywords

Open Source Software, Big Fault Data, Discriminant Analysis, Open Source Project

1. Introduction

Many open source software (OSS) are used in various areas of mobile devices, IoT, server-side application, cloud computing, edge computing and database software. The development paradigm of OSS is different from the typical soft-

ware development style. In particular, the maintenance phase is the fault-data-driven fixing style by using the bug tracking system. The big fault data sets are recorded on the OSS bug tracking system. Then, the size of fault data is approximately over tens of thousands lines. However, it is very difficult to assess the OSS reliability by using the typical statistical method, because the size of data recorded on the bug tracking system is very large scale. As an example, there is the problem for the degrees of freedom in case of the statistical approach. Generally, the degree of freedom is the number of data. Therefore, it is very difficult to decide the degrees of freedom in case of the big data. In this paper, we propose the fusion-method of statistical and stochastic modeling approaches.

The traditional reliability assessment methods based on software reliability growth models have been proposed by several research groups [1] [2]. Moreover, several research papers for OSS reliability assessment have been published in the past [3] [4] [5]. On the other hand, few statistical methods for OSS reliability assessment have been proposed [6] [7], because it is difficult to assess by using the statistical analysis.

We propose the fusion-method project assessment based on quantification method of second type and jump diffusion process model. Then, we can resolve the statistical problem in case of the large scale data analysis such as the big fault data by using our method. Moreover, we show several analysis examples based on the proposed method by using the actual big fault data.

We show the organization of this paper. First, Section 2 proposes the linear discriminant analysis for big fault data. Then, several categorical data are analyzed by using the actual fault data. Moreover, we discuss the jump diffusion process model as the stochastic approach. Section 3 shows the fusion-method of the statistical and stochastic modeling approaches by using the actual OSS fault data. Section 4 summarizes the characteristics of our method.

2. Statistical and Stochastic Modeling Approaches

In terms of the jump term in the jump diffusion process, it is difficult to estimate the unknown parameters of jump term, because the jump diffusion process model has the different stochastic processes. In particular, the jump diffusion process model has two stochastic processes consisting of the Wiener process and jump diffusion one. Also, the jump diffusion process model is assumed that the Wiener process is independent of jump diffusion process. Therefore, we can define the parameter of jump term individually. Then, we propose the estimation method of jump term parameters by using the linear discriminant analysis according to the following procedure.

Step 1: Generally, it is difficult to understand the total trends of big fault data. Therefore, we apply the linear discriminant analysis in order to confirm the correlation of the specified factor for all factor. Thereby, we can understand the mutual interaction in the big fault data.

Step 2: Then, we focus on the contribution rate based on the analysis results

by linear discriminant analysis. In particular, the contribution rate is the important measure, because the contribution rate means the changing rate of each factor for changes in the entire data. Therefore, we consider that the estimates of jump term parameters by using the contribution rate will be useful to assess the reliability considering the characteristic of big fault data.

Step 3: We estimate the mean and variance of contribution rates for all factors from the analysis results of big fault data. Then, we apply the mean and variance of contribution rates to the unknown parameters of jump term.

Step 4: Then, we can show several reliability assessment measures based on the jump diffusion process model.

2.1. Linear Discriminant Analysis

We focus on Fisher's linear discriminant analysis. Considering the linear discriminant analysis, it is assumed that the applied data is satisfied the following conditions:

1) The data is based on the normal distribution.

2) Each class has the same covariance matrix.

3) Variables are independent each other.

We show analysis examples by using the Apache HTTP Server Project [8] as the OSS. At first, we analyze the potential of normal distribution for actual data sets. We show the approximate curve based on normal distribution for all category in actual fault big data in **Figures 1-3**. The number of 10,000 lines fault data are included in **Figures 1-3**, respectively. Then, the number of whole data is about 130,000 category data. The X-axes of **Figures 1-3** mean the number of appearance for each factor. The Y-axes of **Figures 1-3** show the occurrence rate for each factor. Variable software reliability assessment methods based on the stochastic models have been proposed in the past. Almost all of these methods of software reliability assessment are based on the data in terms of the number of software faults. Therefore, we focus on the data in terms of the number of faults. On the whole, we found that almost data factors are approximately based on the normal distribution from **Figures 1-3**.

We show the numerical examples based on linear discriminant analysis in **Figures 4-13**, respectively. In particular, we discuss the estimation results of **Figures 4-13** as follows:

Product: The level of uniformity is high. However, the data has two factors only.

Component: Two clusters are structured by calculation. In particular, the group with core component and main component is placed to left side, the right side group is the small size component such as Other and sub-component.

Version: As with Component, two clusters are structured by the analyzation. The right side cluster becomes large. In particular, the faults group of newly version is placed to the right side cluster.

Reporter: Four clusters are composed by the analyzation. We can consider

that this is the unbiased result. This software has been reported by various un-uniformed reporters.

Severity: Two clusters are estimated. In particular, the right-bottom cluster becomes large. The clusters of Reporter and Severity may be the same situation, because two clusters are the same shape.

Status: We cannot find the characteristics from this figure.

Resolution: There are three types of cluster.

Hardware: There are two types of cluster. In particular, we found that the Reporter, Severity, and Hardware show the same tendency.

OS: The specified factor has biased.

Summary: The level of uniformity is high.

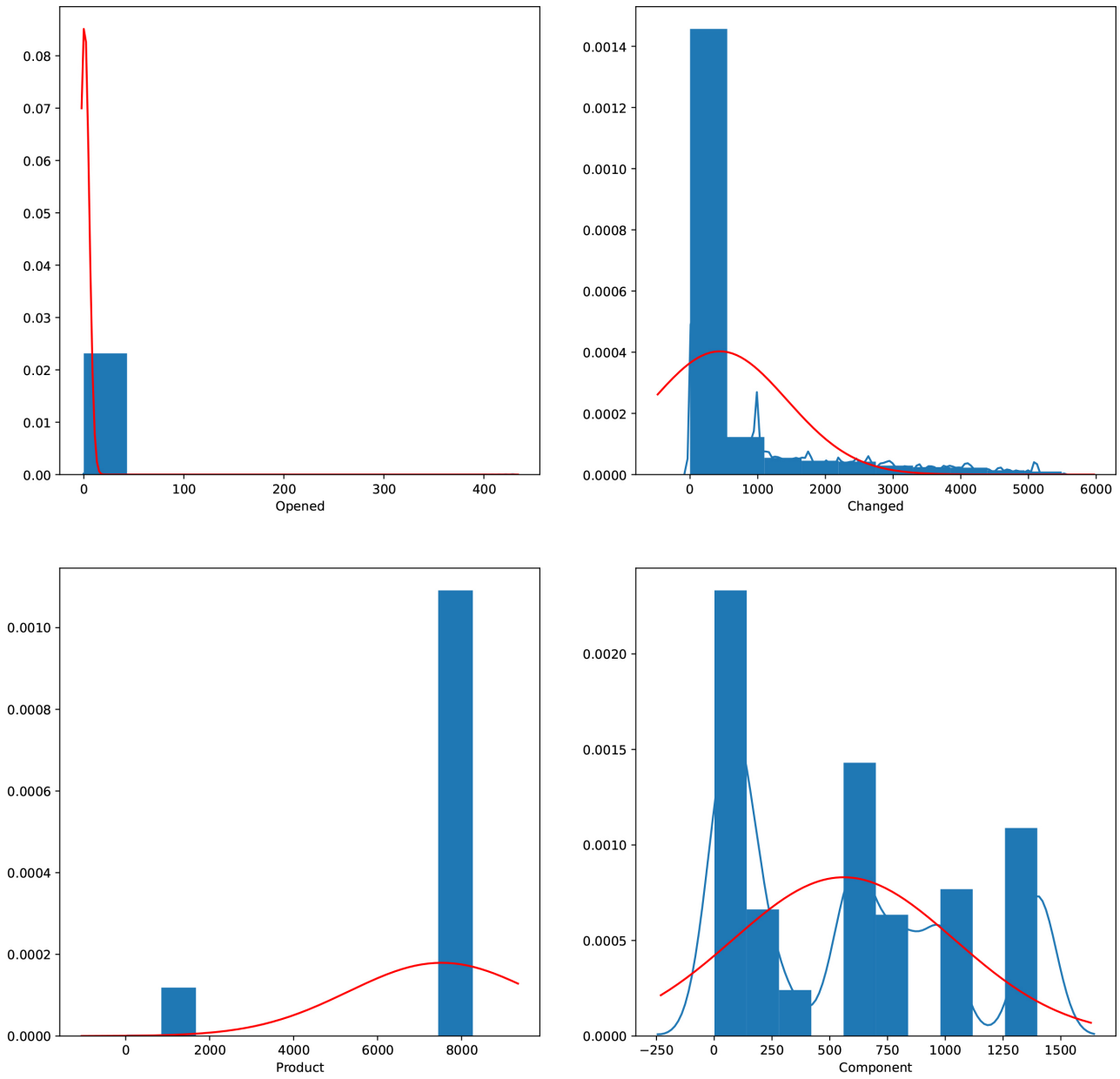


Figure 1. The approximate curve based on normal distribution for 1st category in actual fault big data.

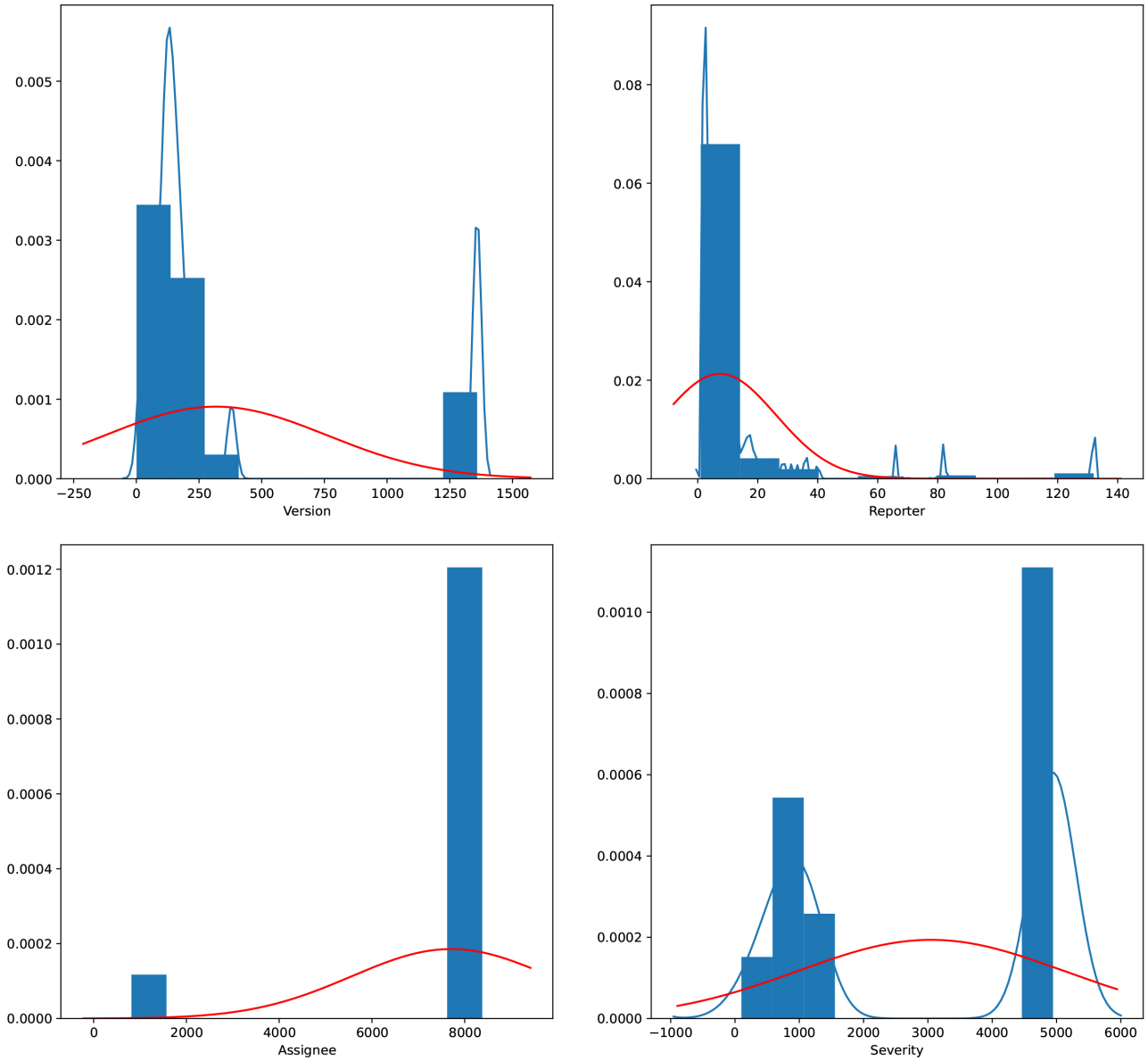


Figure 2. The approximate curve based on normal distribution for 2nd category in actual fault big data.

In this paper, we analysis the highest contribution rate for all factors, because the contribution rate means the changing rate of each factor for changes in the entire data. Table 1 shows the estimated largest contribution rate for each factor. From Table 1, we found that the mean is 0.19294, the unbiased standard deviation is 0.05401 in case of the minimum value. In this paper, we can define the jump term of jump diffusion process model by using the estimated mean and unbiased standard deviation obtained from the contribution rate. We consider that the degrees of influence for the number of faults becomes large in case of the maximum value of contribution rates. On the other hand, the degrees of influence for the number of faults becomes small in case of the minimum value of contribution rates, *i.e.*, it is appropriate to assess by using the jump term of jump diffusion process model.

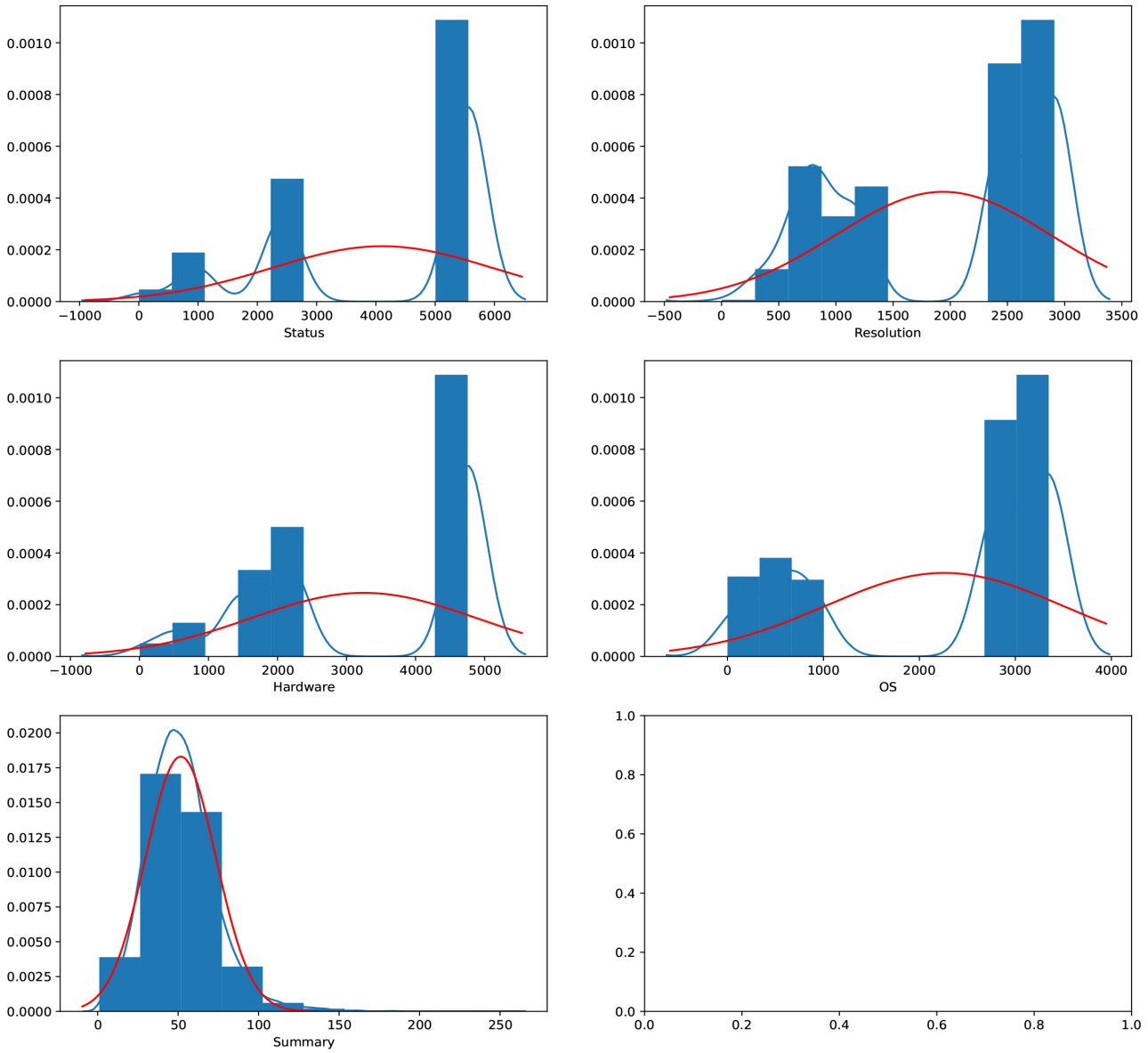


Figure 3. The approximate curve based on normal distribution for 3rd category in actual fault big data.

Table 1. The estimated of contribution rates.

	Maximum value	Minimum value
Product	0.82721	0.17279
Component	0.59082	0.00110
Version	0.87069	0.00088
Reporter	0.52753	0.00069
Severity	0.56814	0.00366
Status	0.72707	0.00088
Resolution	0.63607	0.00010
Hardware	0.75804	0.00084
OS	0.77949	0.00026
Summary	0.20772	0.02588
Mean	0.6493	0.19294
Unbiased SD	0.02071	0.05401

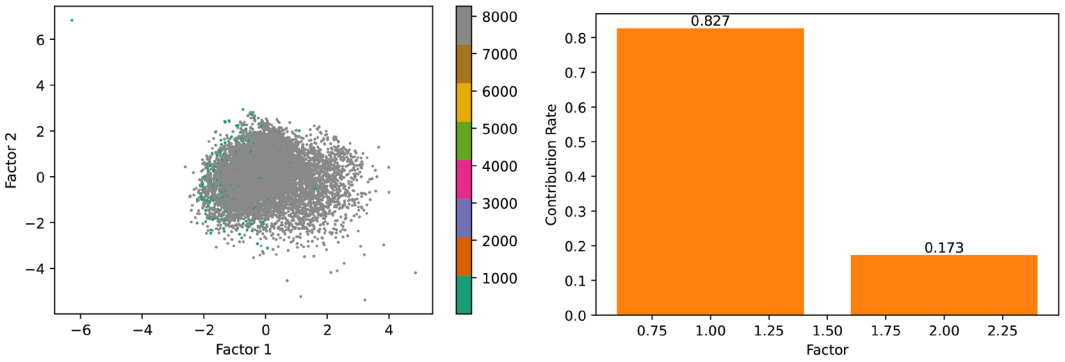


Figure 4. The estimate based on linear discriminant analysis for Product of actual fault big data.

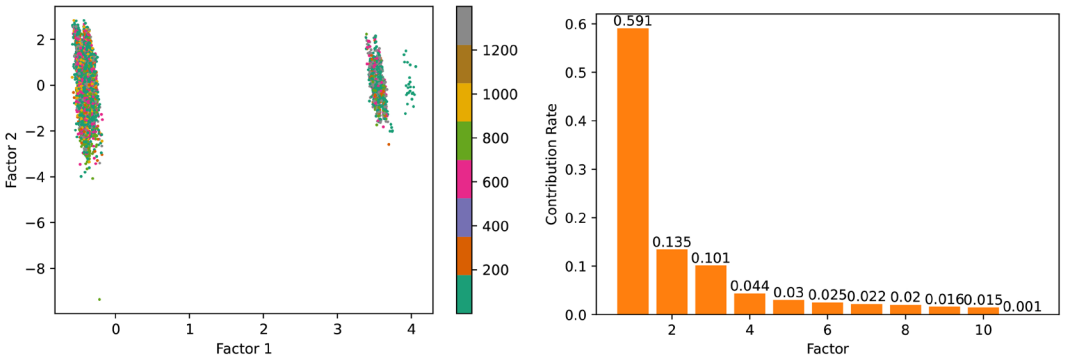


Figure 5. The estimate based on linear discriminant analysis for Component of actual fault big data.

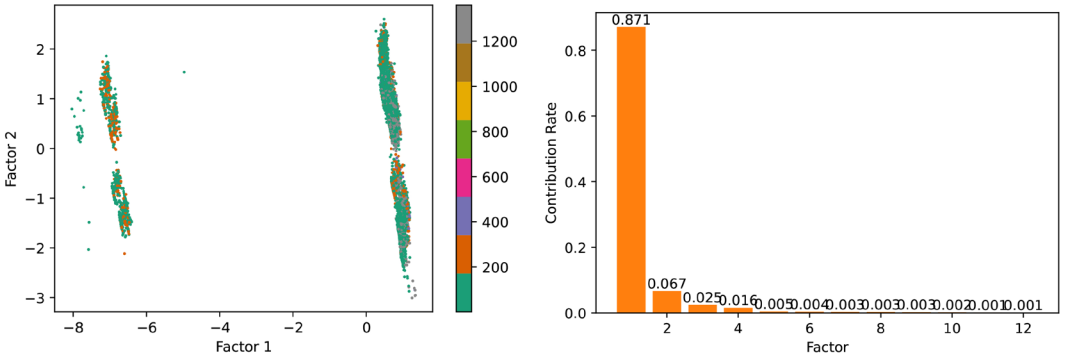


Figure 6. The estimate based on linear discriminant analysis for Version of actual fault big data.

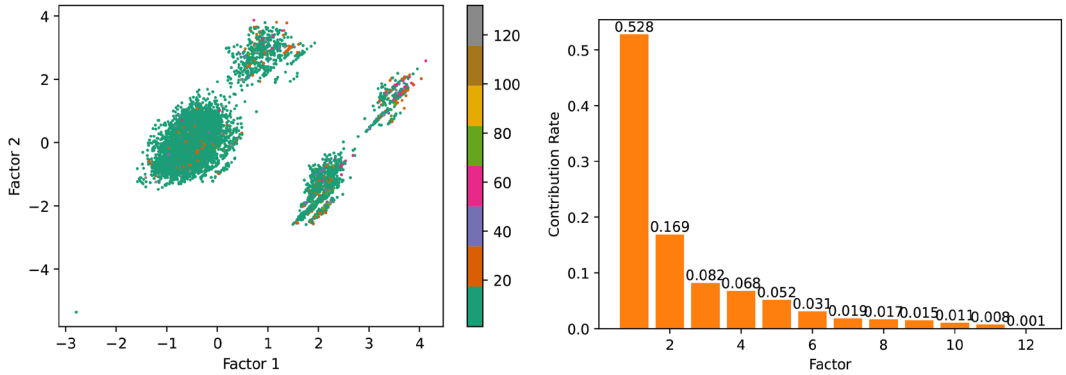


Figure 7. The estimate based on linear discriminant analysis for Reporter of actual fault big data.

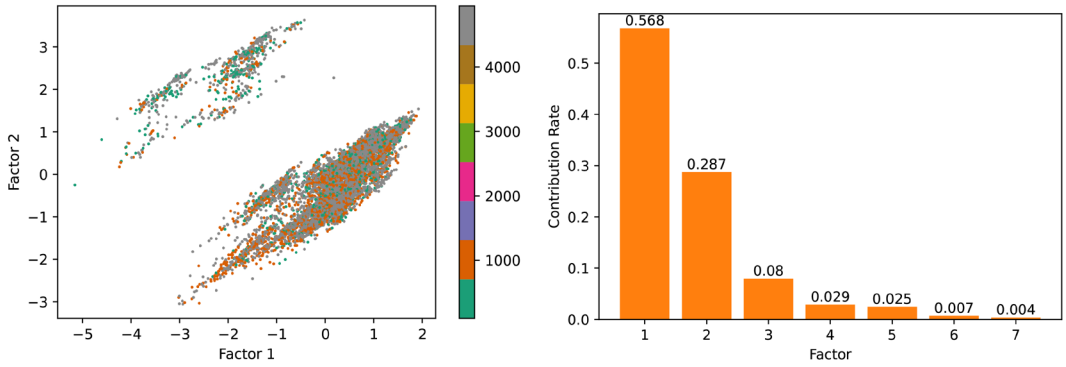


Figure 8. The estimate based on linear discriminant analysis for Severity of actual fault big data.

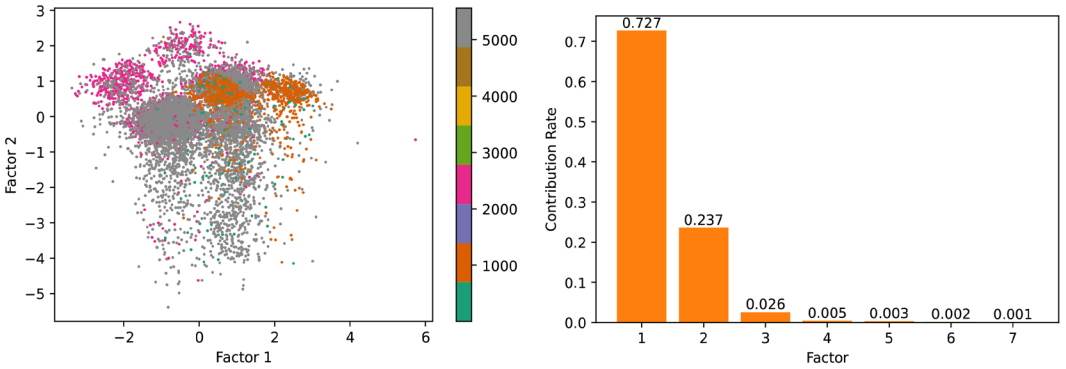


Figure 9. The estimate based on linear discriminant analysis for Status of actual fault big data.

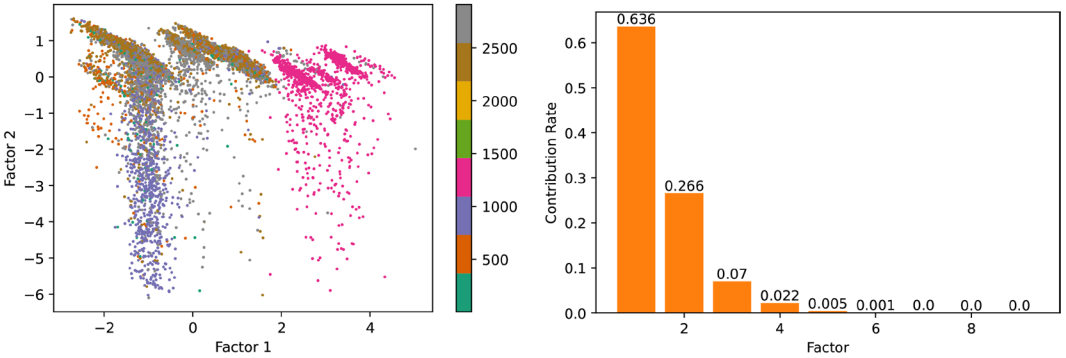


Figure 10. The estimate based on linear discriminant analysis for Resolution of actual fault big data.

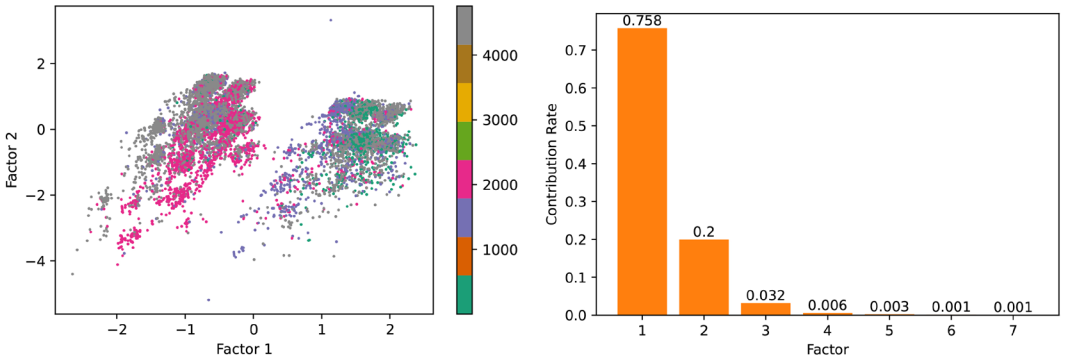


Figure 11. The estimate based on linear discriminant analysis for Hardware of actual fault big data.

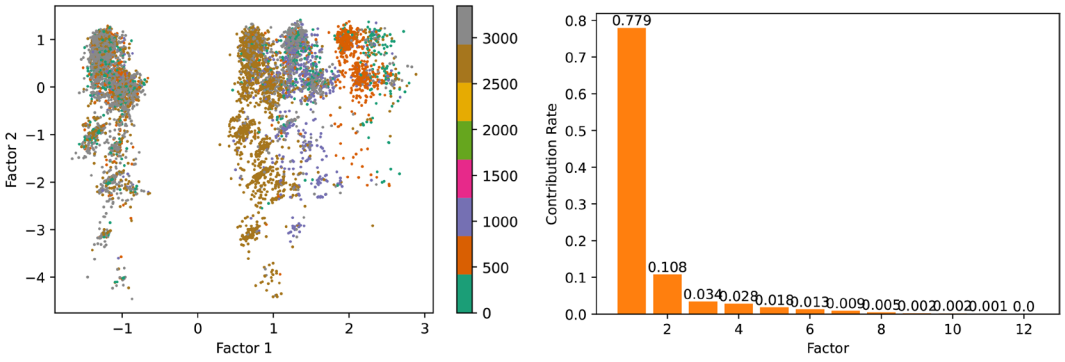


Figure 12. The estimate based on linear discriminant analysis for OS of actual fault big data.

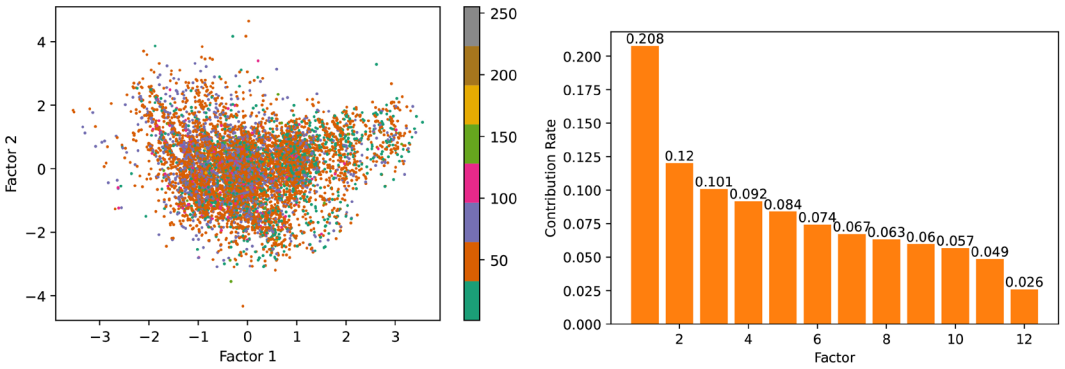


Figure 13. The estimate based on linear discriminant analysis for Summary of actual fault big data.

2.2. Jump Diffusion Process Model

We apply a stochastic differential equation model to manage the maintenance effort in the operational phase of OSS projects. In the past, our research group has been proposed the jump diffusion process model [9] [10]. First, we discuss the flexible jump diffusion process model. The jump diffusion process model has derived from the following stochastic differential equation with Brownian motion [11] [12]:

$$\frac{\mathrm{d}J(t)}{\mathrm{d}t} = \{D(t) + \gamma g(t)\}\{p - J(t)\}. \tag{1}$$

The parameters of Equation (1) are as follows:

$J(t)$: the cumulative maintenance effort expenditures up to operational time $t(t \geq 0)$ in the OSS development project, this takes on continuous real values.

$D(t)$: the increase rate of maintenance effort at operational time t and a non-negative function,

γ: a positive constant representing a magnitude of the irregular fluctuation,

$g(t)$: a standardized Gaussian white noise,

p: the estimated amount of maintenance effort required until the end of operation.

We extend to the following stochastic differential equation of an Itô type [11]:

$$\mathrm{d}J(t) = \left\{D(t) - \frac{1}{2}\gamma^2\right\}\{p - J(t)\}\mathrm{d}t + \gamma\{p - J(t)\}\mathrm{d}\omega(t). \tag{2}$$

Then, the parameter $\omega(t)$ is defined as

$\omega(t)$: one-dimensional Wiener process which is formally defined as an integration of the white noise $g(t)$ with respect to time *t*.

Then, the jump term can be added to the stochastic differential equation models in order to incorporate the irregular state around the time t by various external factors in the operation phase of OSS project. Then, the jump-diffusion process [9] [10] [13] is given as

$$\mathrm{d}J_j(t) = \left\{D(t) - \frac{1}{2}\gamma^2\right\}\left\{p - J(t)\right\}\mathrm{d}t + \gamma\left\{p - J_j(t)\right\}\mathrm{d}\omega(t) + d\left\{\sum_{i=1}^{\nu_t(\lambda)}(\rho_i - 1)\right\}. \quad (3)$$

$\nu_t(\lambda)$: a Poisson point process with parameter λ at operation time *t*. The number of occurred jumps, and λ the jump rate. $\nu_t(\lambda)$ and $\omega(t)$, and ρ_i are assumed to be mutually independent.

ρ_i: *i*-th jump range.

By using Itô's formula [11] [12], the solution of the former equation can be obtained as follows:

$$J_{je}(t) = p\left[1 - \exp\left\{-qt - \gamma\omega(t) - \sum_{i=1}^{\nu_t(\lambda)}\log\rho_i\right\}\right], \quad (4)$$

$$J_{js}(t) = p\left[1 - (1+qt)\exp\left\{-qt - \gamma\omega(t) - \sum_{i=1}^{\nu_t(\lambda)}\log\rho_i\right\}\right]. \quad (5)$$

Considering the effort expenditure phenomenon, we define the normal distribution function as Gaussian Jump-diffusion process in order to consider the characteristics of software effort-growth phenomena:

$$\rho_i \equiv f_i(x) = \frac{1}{\sqrt{2\pi\tau}}\exp\left[-\frac{(x-\mu)^2}{2\tau^2}\right]. \quad (6)$$

Then, we assume that the *i*-th jump range ρ_i are approximately is estimated as the positive values in almost all cases, because the mean value μ keep a large value. The jump process is mutually independent from Wiener process in our model. Then, we will be able to estimate several parameters of jump term separated from ones of Wiener term.

In particular, we apply the mean and unbiased standard deviation obtained from the analysis results of contribution rate in section 2.1 to the parameters μ and τ included in ρ_i, *i.e.*, the estimated mean is 0.6493, the estimated unbiased standard deviation 0.19294 in case of section 2.1.

3. Numerical Examples for Jump Diffusion Process Model

Based on section 2.1, we show several numerical examples for jump diffusion process model. **Figure 14** and **Figure 15** show the estimated sample path cumulative software effort and sample path of the required effort expense for exponential type model. In particular, the jump range, the number of occurred jumps, and the jump rate increase according to the operation procedures go on.

On the other hand, Figure 16 and Figure 17 show the estimated sample path cumulative software effort and sample path of the required effort expense for S-shaped type model. From Figures 14-17, the S-shaped type model is optimistically estimated in comparison with the exponential type model, because the estimated sample path of the required effort expense for S-shaped type model is smaller than the exponential type model in Figure 15 and Figure 17, respectively.

In particular, Figure 18 shows the estimated distribution function $f_i(x)$ in Equation (6) based on the contribution rates. Figure 18 is important role for the proposed method, because the jump parameter is estimated by using the linear discriminant analysis in order to summarize the interaction among complex categories recorded on the big fault data.

The characteristics of our method can estimate the software effort based on several fault category recorded on the bug tracking system. Then, the proposed method can provide the information of mutual interaction among several fault category by using the jump noise. Thereby, the OSS managers will be able to assess the stability of OSS project.

4. Conclusions

This paper has proposed the reliability assessment method based on quantification method of the second type and jump diffusion process model for OSS big fault data. The purposes of the proposed method are as follows:

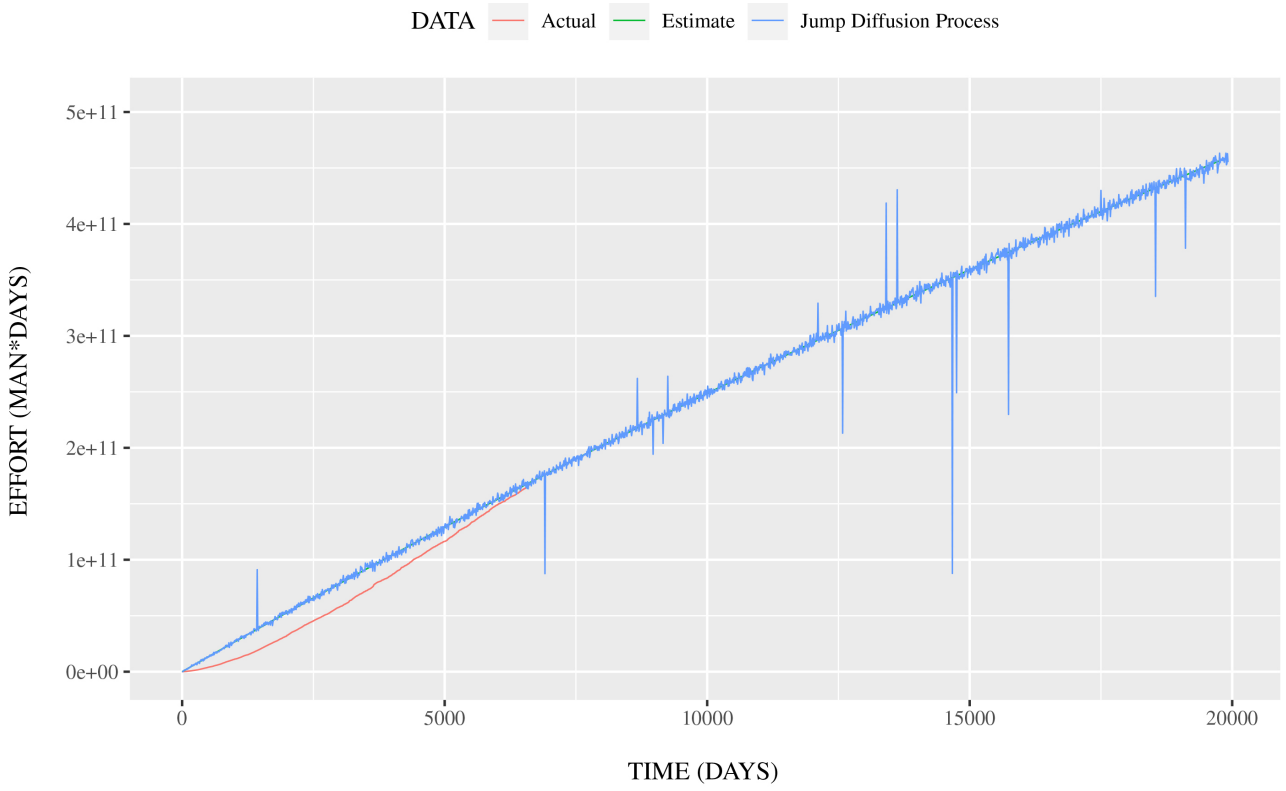


Figure 14. The estimated sample path cumulative software effort for exponential type model.

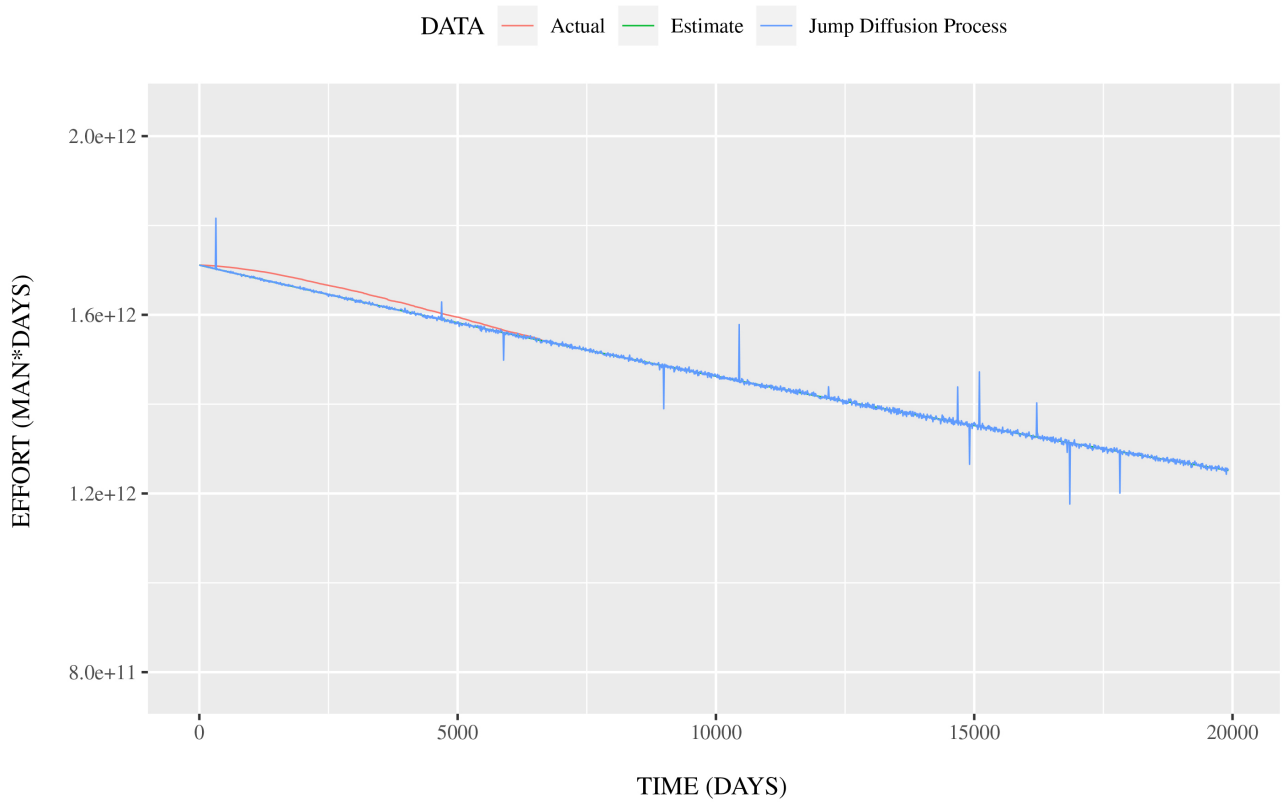


Figure 15. The estimated sample path of the required effort expense for exponential type model.

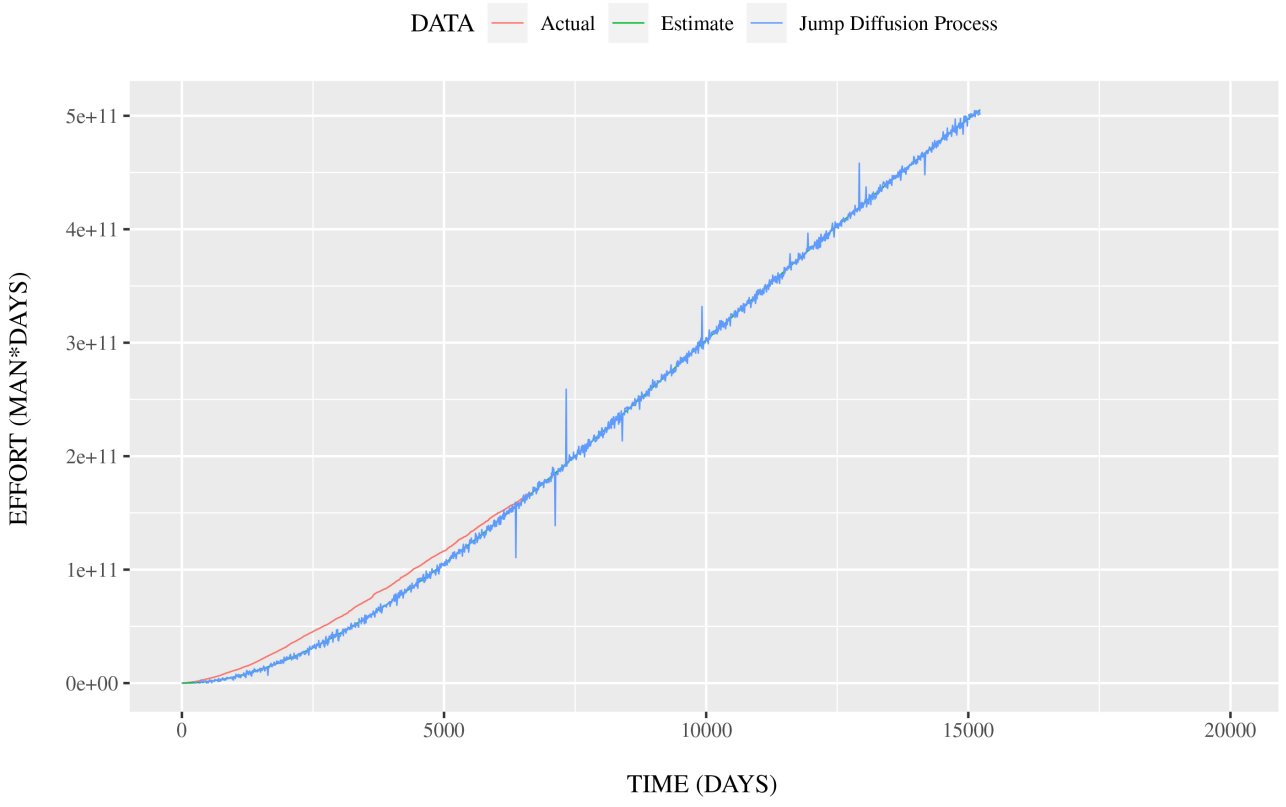


Figure 16. The estimated sample path cumulative software effort for S-shaped type model.

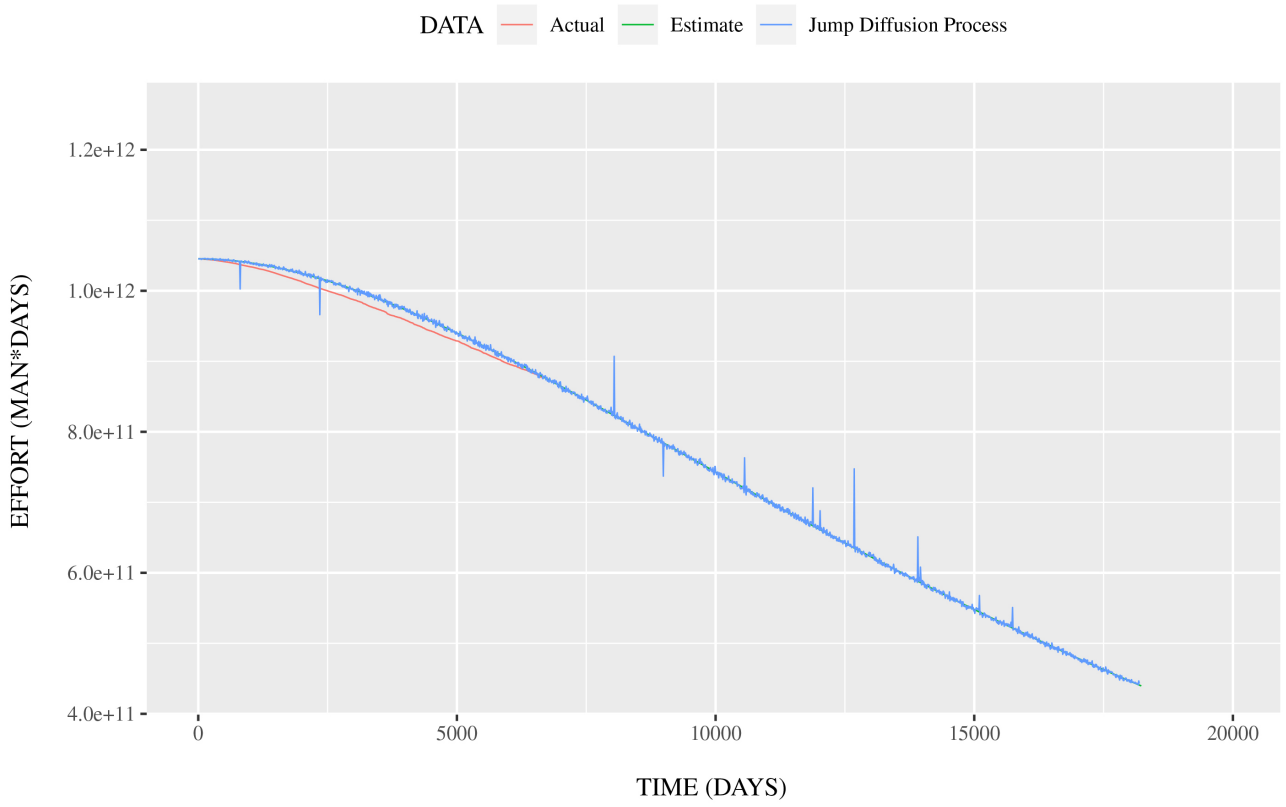


Figure 17. The estimated sample path of the required effort expense for S-shaped type model.

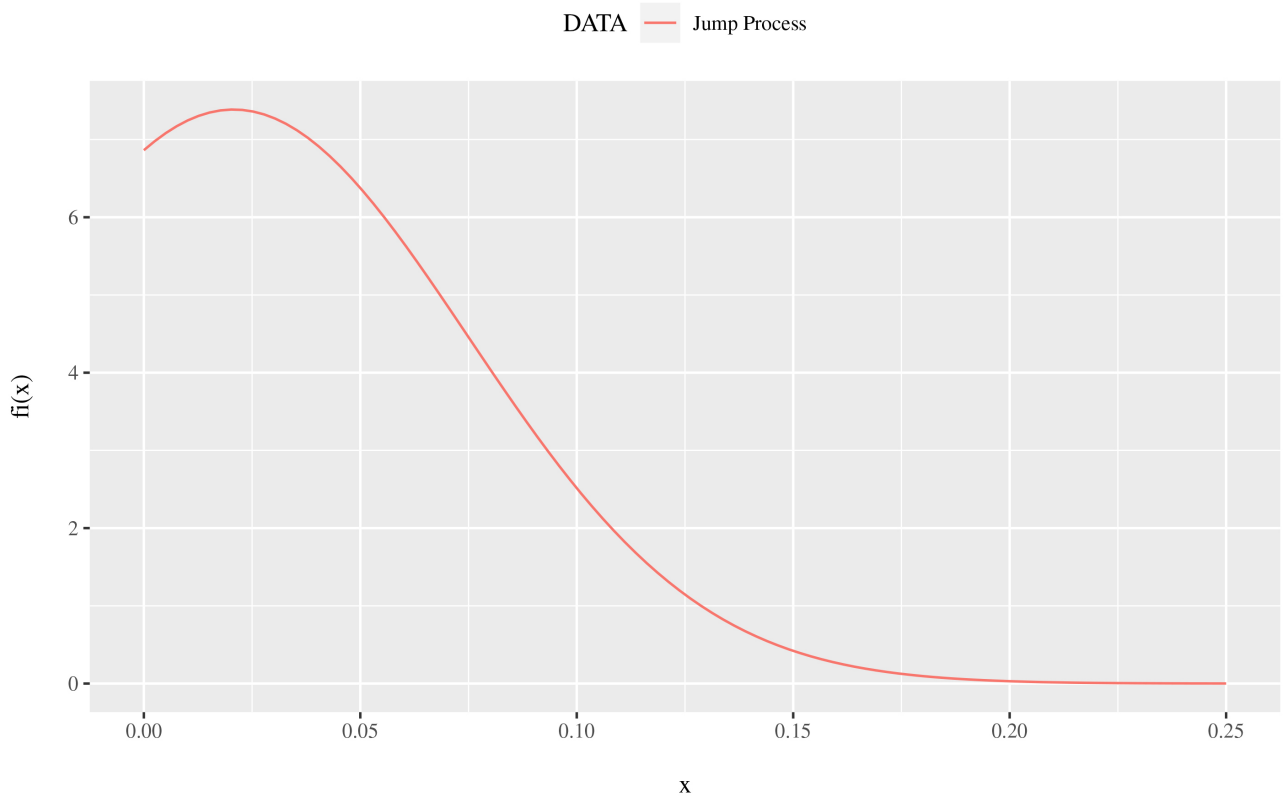


Figure 18. The estimated distribution function $f_i(x)$ in Equation (6) based on the contribution rates.

1) In terms of the quantification method of the second type, it is important to understand several fault categories, because the fault big data sets are recorded with many fault contents. Also, it will be helpful to use many fault contents, not only effort data. Then, the contribution rate is very important measure. The fault category has the large impact, if the value of contribution rate is large. On the other hand, the fault category has the small impact, if the value of contribution rate is small. In particular, the factor in case the small value of contribution rate has little effect on the software effort. This means that the factors in case the small value of contribution rate will appear as the noise for software effort.

2) In terms of the jump diffusion process model, we can understand the unexpected changes by using the jump term of jump diffusion process model. However, it is difficult to estimate the parameters of jump term in terms of fault big data because of the complex category data. Therefore, we have proposed the estimation method by using the linear discriminant analysis as known the quantification method of the second type. Thereby, it is possible to assess considering the standpoint of the interaction among several fault category.

Above mentioned reasons, the proposed method will be useful to assess the OSS development effort by using the jump noises from the standpoint of the interaction among several fault factors. Therefore, our method can simply use for the other OSS. The proposed method can find the main factors as explanatory variables affecting the quality control. Thereby, the OSS developer will be able to easily assess the quality from the standpoint of the condition recorded from actual fault big data.

Acknowledgements

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

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Yamada, S. (2014) Software Reliability Modeling: Fundamentals and Applications. Springer-Verlag, Tokyo/Heidelberg. https://doi.org/10.1007/978-4-431-54565-1_1

[2] Kapur, P.K., Pham, H., Gupta, A. and Jha, P.C. (2011) Software Reliability Assessment with OR Applications. Springer-Verlag, London. https://doi.org/10.1007/978-0-85729-204-9

[3] Yamada, S. and Tamura, Y. (2016) OSS Reliability Measurement and Assessment. Springer International Publishing, Switzerland. https://doi.org/10.1007/978-3-319-31818-9

[4] Norris, J. (2004) Mission-Critical Development with Open Source Software. *IEEE Software Magazine*, **21**, 42-49. https://doi.org/10.1109/MS.2004.1259211

[5] Singh, V.B., Sharma, M. and Pham, H. (2017) Entropy Based Software Reliability

Analysis of Multi-Version Open Source Software. *IEEE Transactions on Software Engineering*, 1207-1223. https://doi.org/10.1109/TSE.2017.2766070

[6] Rahmani, C., Azadmanesh, A. and Lotfi, N. (2010) A Comparative Analysis of Open Source Software Reliability. *Journal of Software*, **5**, 1384-1394. https://doi.org/10.4304/jsw.5.12.1384-1394

[7] Nagaraju, V., Shekar, V., Steakelum, J., Luperon, M., Shi, Y. and Fiondella, L. (2019) Practical Software Reliability Engineering with the Software Failure and Reliability Assessment Tool (SFRAT). *SoftwareX*, **10**, 1-6. https://doi.org/10.1016/j.softx.2019.100357

[8] The Apache Software Foundation, The Apache HTTP Server Project. http://httpd.apache.org/

[9] Tamura, Y., Sone, H. and Yamada, S. (2019) Productivity Assessment Based on Jump Diffusion Model considering the Effort Management for OSS Project. *International Journal of Reliability, Quality and Safety Engineering* (World Scientific), **26**, 1950022. https://doi.org/10.1142/S0218539319500220

[10] Tamura, Y., Sone, H., Sugisaki, K. and Yamada, S. (2019) A Method of Parameter Estimation in Flexible Jump Diffusion Process Models for Open Source Maintenance Effort Management. *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, Macau, 15-18 December 2019, CD-ROM (Reliability and Maintenance Engineering 2). https://doi.org/10.1109/IEEM44572.2019.8978611

[11] Arnold, L. (1974) Stochastic Differential Equations—Theory and Applications. John Wiley & Sons, New York.

[12] Yamada, S., Kimura, M., Tanaka, H. and Osaki, S. (1994) Software Reliability Measurement and Assessment with Stochastic Differential Equations. *IEICE Transactions on Fundamentals*, **E77-A**, 109-116.

[13] Merton, R.C. (1976) Option Pricing When Underlying Stock Returns Are Discontinuous. *Journal of Financial Economics*, **3**, 125-144. https://doi.org/10.1016/0304-405X(76)90022-2