Reliability and Maintainability Analysis and its ToolBased on Deep Learning for Fault Big Data

Yoshinobu Tamura[1], Shigeru Yamada[2]

[1]*Tokyo City University*
Tamazutsumi1-28-1, Setagaya-ku, Tokyo 158-8557, Japan
[1]*tamuray@tcu.ac.jp*
[2]*Tottori University*
Minami 4-101, Koyama, Tottori-shi, 680-8552, Japan
[2]*yamada@tottori-u.ac.jp*

***Abstract* — Recently, many fault data sets are recorded on the bug tracking systems. These bug tracking systems are used on the Website for such as open source project management. The software managers of open source projects can comprehend the fault status by using the bug tracking system. However, these software systems for development support such as the bug tracking system cannot estimate the future trend of fault data. We discuss the methods of reliability and maintainability assessment based on the deep learning for fault big data. In particular, we develop the reliability and maintainability analysis tool based on deep learning for fault big data by using the latest programing technology. Moreover, we show several numerical illustrations of the developed software tool by using the fault big data in the actual software projects.**

***Keywords* — Software tool, deep learning, software reliability, maintainability, fault big data**

I. INTRODUCTION

Several software reliability models[1-3] have been used to assess the reliability for quality management under the system testing phase of waterfall model in software development. On the other hand, it is difficult to assess the reliability of open source software (OSS), because of the development style of OSS is different from the typical existing software development paradigm.At present, several software reliability growth models for OSS have been proposed[4]. However, it is difficult for the software managers to select the optimal software reliability model for the actual software development project.

Considering the OSS development paradigm, the bug tracking systems are used in many OSS projects. In the bug tracking systems, many fault data sets are recorded by several users and developers. It will be helpful for OSS project managers to assess the reliability and maintainability of OSS,if many fault data recorded on the bug tracking system are used for software quality improvement. However, many software reliability models cannot use many fault data sets recorded on the bug tracking system because of the model parameter constraints. Then, it will be able to assess OSS reliability and maintainability based on many fault data setsby using the nonparametric method such as deep learning.

In this paper, we focus on the OSS reliability and maintainability assessment based on the deep learning by using the OSS fault big data. Then, we discuss the method of OSS reliability and maintainability assessment based on deep learning. In particular, we develop the application software for the reliability and maintainability assessment based on the deep learning. Then, the developed application software is used by the latest technologies such as NW.js, statistical computing R, HTML, JavaScript, CSS3, ggplot2 and plotly libraries of R. Also, several numerical illustrations of software reliability and maintainability assessment tool by using the fault data in the actual OSS projects are shown in this paper. Moreover, we discuss the results of numerical illustrations by using our application software.

II. RELIABILITY AND MAINTAINABILITY ASSESSMENT BASED ON DEEP LEARNING

The structure of the deep learning in this paper is shown in Fig. 1. In Fig. 1, $z_l(l = 1, 2, \cdots, L)$ and$z_m(m = 1, 2, \cdots, M)$ means the pre-training units. Also, $o_n(n = 1, 2, \cdots, N)$ is the amount of compressed characteristics. Several algorithms in terms of deep learning have been proposed[5-10]. In this paper, we apply the deep neural network to learn the fault data on bug tracking systems of open source projects.We apply the following amount of information to the parameters of pre-training units. Then, the objective variable is given as the levels of software reliability trend as shown in Table 1.

We apply 2 kinds ofreliability trend level as the coefficient value to the amount of compressed characteristics, i.e., Positiveand Negative, respectively. The following 9 kinds of explanatory variables are set to the amount of pre-training units:

- Date recorded on bug tracking system
- Name of software product
- Name of software component
- Number of software version

978-1-5090-3012-5/17/$31.00 ©2017 IEEE

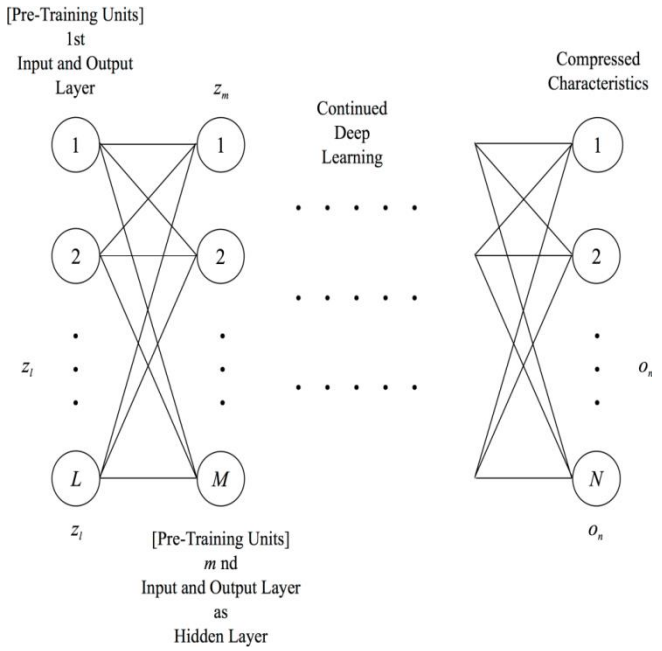

Fig. 1The structure of deep learning in this paper.

- Nickname of fault reporter
- Nickname of fault assignee
- Status of software fault
- Name of operating system
- Fault severity level

Then, the data of each explanatory variable are converted from the character data to the numerical values such as the rate of occurrence.

The mean time between software failures(MTBF) is useful to measure theproperty of the frequency of software failure-occurrence.We definethe measure of MTBF as follows:

$$t_k = t_{k-1} + dl_{k-1} \cdot SD_{k-1}, \qquad (1)$$

where t_k is k-th MTBF, dl_k means the estimated k-th index number as the coefficient value based on deep learning, e.g., the index number "1" means the reliability growth trend, the index number "-1" means the reliability regression trend. Also,SD_k means k-thstandard deviation as given by the following equation:

$$SD_k = \sqrt{\frac{1}{k-1}\sum_{n=1}^{k}\left(t_n - \frac{1}{n}\sum_{l=1}^{n} t_l\right)^2}. \qquad (2)$$

Similarly, the correctiontime between software failures(CTBF) is useful to measure theproperty of the frequency of software maintenance. We define the correction time between software failures (CTBF) as follows:

TABLE I
The Coefficient Values for Reliability Trends.

Reliability Trends	Coefficient Value
Positive	Increase (Index number 1)
Negative	Decrease (Index number -1)

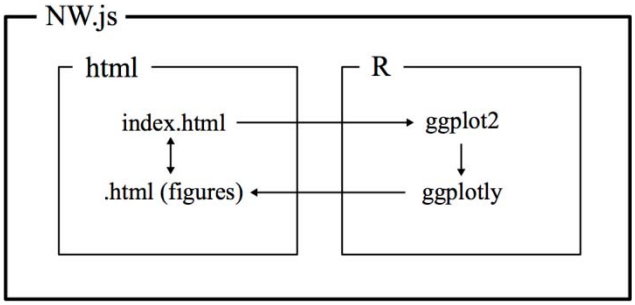

Fig. 2 Theframework of deep learning in this paper.

$$c_k = c_{k-1} + dl_{k-1} \cdot SD_{k-1}, \qquad (3)$$

wheret_k is k-th CTBF, anddl_k means the estimated k-th index number as the coefficient value based on correction time by using the deep learning. Also, SD_k means k-thstandard deviation as given by the following equation:

$$SD_k = \sqrt{\frac{1}{k-1}\sum_{n=1}^{k}\left(c_n - \frac{1}{n}\sum_{l=1}^{n} c_l\right)^2}. \qquad (4)$$

Development of Application Software for Reliability and Maintainability Analysis

The specification requirement of the reliability and maintainability analysistool based on deep learning for fault big data are shown as follows:

1. This tool should be operated by clicking the mouse button and typing on the keyboard to input the data through GUIsystem. In particular, theuser experience design is adopted as the important element of our tool.

2. The developed applicationsoftware should be implemented by NW.js[11], statistical computing R[12], HTML, JavaScript, CSS3, ggplot2, and plotly[13] as the latest technologies. This tool is developed as a native application on Windows, Unix, and macOSoperating system. Also, this tool operates as Webapplication.

3. The method of deep learning discussed in chapter II is used as thecoefficient valuedl_k in the proposed method.

4. The application software illustrate the estimated MTBF and CTBF based on the estimatedcoefficient valuedl_k. Then, the MTBF and CTBF based on10-fault moving-average and 50-fault moving-average are used inour tool.

In particular, the developed software application is implemented by using the structure as shown in Fig. 2. For examples, thesimplified source code of developed application software is shown as follows:

```
----- index.html -----
<a href="javascript:void(0);" onclick="mtbfc();">MTBF</a>
<a href="javascript:void(0);" onclick="ctbfc();">CTBF</a>
functionmtbfc() {
var script = document.createElement('script');
script.src = 'mtbfc.js';
document.body.appendChild(script);
  }
functionctbfc() {
var script = document.createElement('script');
script.src = 'ctbfc.js';
document.body.appendChild(script);
  }
```

```
----- ggplot2 -----

library(ggplot2)
mtbf<- ggplot(data=dw,
aes(x, y, geom="line", colour=DATA)) +
xlim(0,length(test_data[,1])) + ylim(-2,3) +
geom_line() + xlab('\nDATA (LINES)') +
ylab('COEFFICIENT VALUE\n') +
theme_bw(base_size=15) + theme(legend.position=c(.16, .87),
legend.title=element_blank(),
panel.border = element_rect(size=1.5, colour="black"),
panel.grid.major = element_line(size=0.4, colour="gray",
linetype="solid"), panel.grid.minor = element_blank())
```

```
----- ggplotly -----

library(plotly)
mtbf<- ggplotly(mtbf)
htmlwidgets::saveWidget(mtbf, "mtbf.html")
```

```
----- *.html (figures) -----
functionmtbf() {
window.open('data/mtbf.html');
  }

functionctbf() {
window.open('data/ctbf.html');
  }
```

III. NUMERICAL ILLUSTRATIONS

In this paper, we focus on Apache HTTP server[14] in order to evaluate the performance of our methods. Several numerical examples by using the data sets for Apache HTTP server as OSS are shown in this section. The data used in this paper are collected in the bug tracking system on the website of Apache HTTP server open source project.

We obtain 10,000 fault data sets from the data recorded on bug tracking system of Apache HTTP server. Then, 90% of the recorded data is used as the learning data.We show themain screen of our software application in Fig. 3. Also, the estimated 10-fault moving-average MTBF based on deep learning by usingtesting data obtained from the bug tracking system in Fig. 4. Similarly, the estimated 10-fault moving-average CTBF based on deep learning by usingtesting data obtained from the bug tracking system in Fig. 5.

Moreover, Fig. 6 shows theestimated 50-fault moving-average MTBF based on deep learning by using the testing data sets.Similarly, Fig. 7shows theestimated 50-fault moving-average CTBF based on deep learning by using the testing data sets.

From Figs. 3 and 4, we foundthat the MTBF and CTBF are useful for the software managers to understand the OSS reliability and maintainability by using our tool.Moreover, we found thatour tool will be helpful for the OSS managers to confirm the trend of reliability and maintainabilityin terms of the long-term prediction from Figs. 6 and 7.

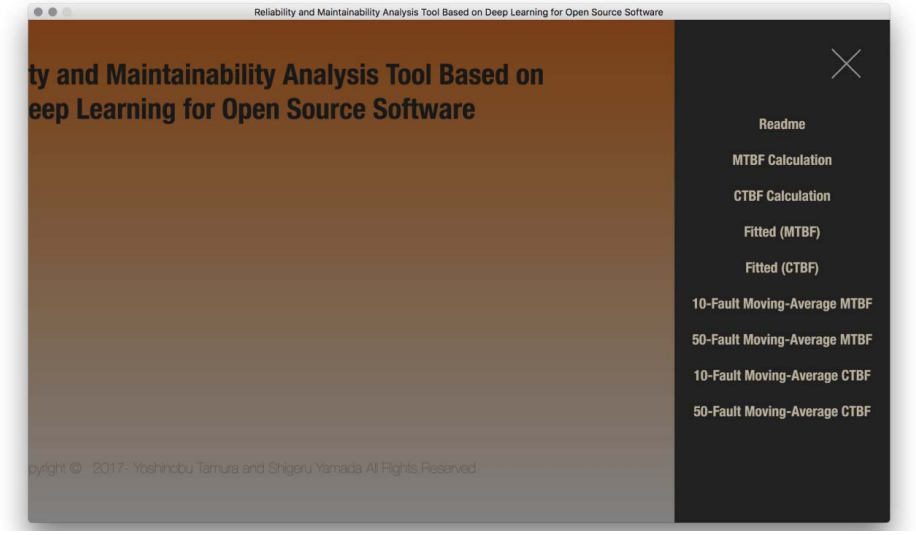

Fig. 3.The main screen of our software application.

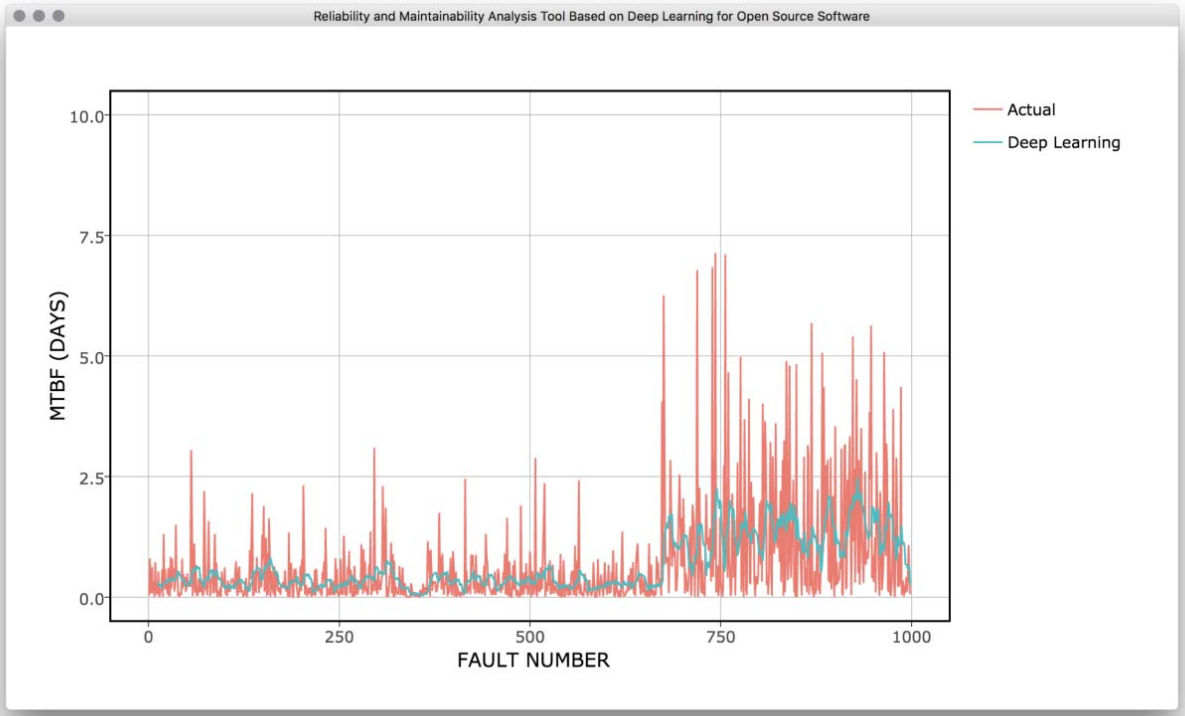

Fig. 4The estimated 10-fault moving-average MTBF based on deep learning.

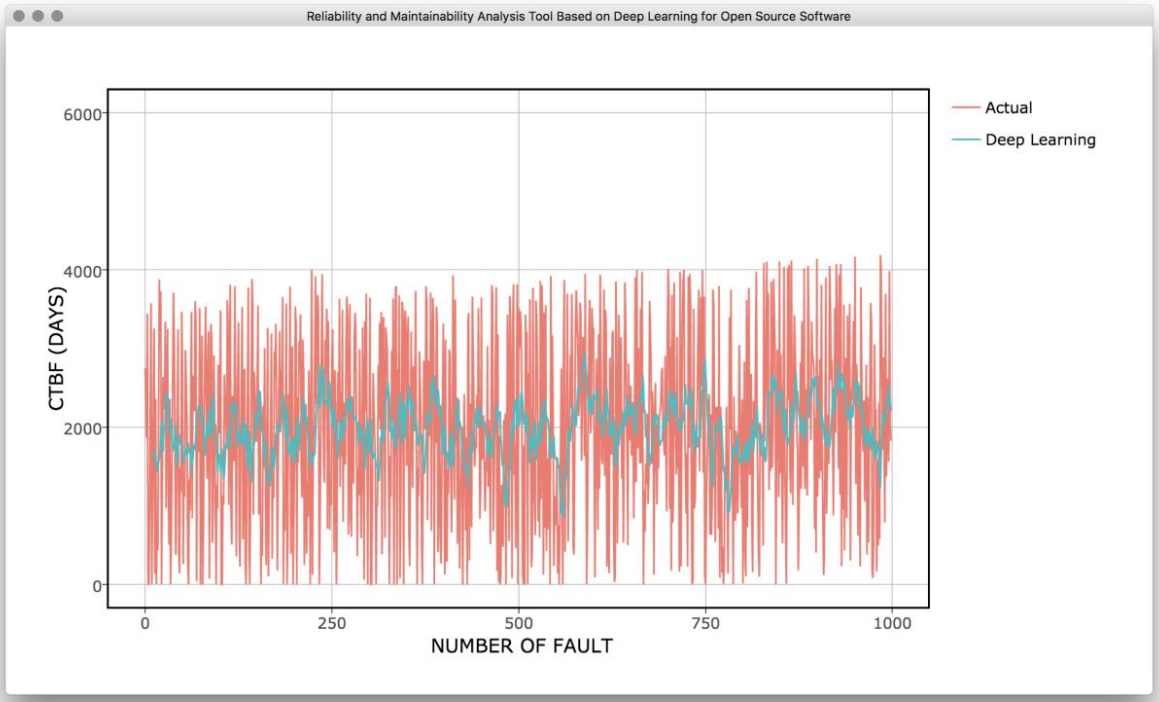

Fig. 5The estimated 10-fault moving-average CTBF based on deep learning.

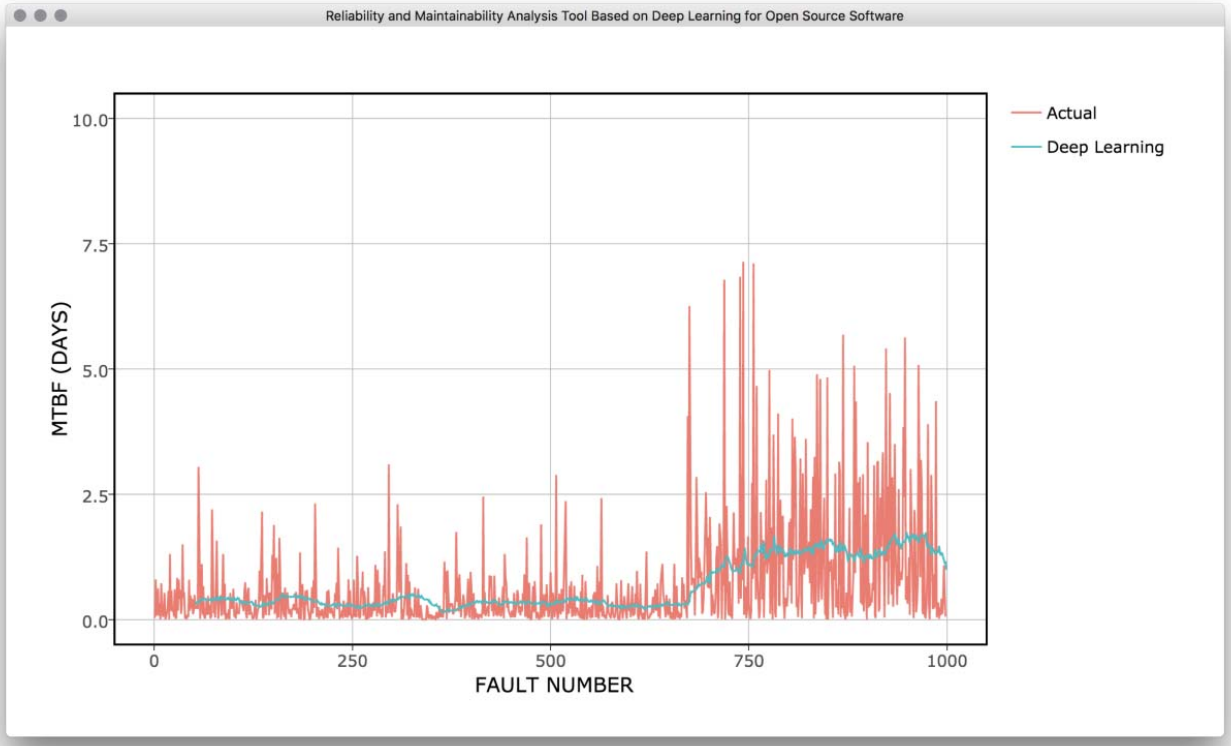

Fig. 6The estimated 50-fault moving-average MTBF based on deep learning.

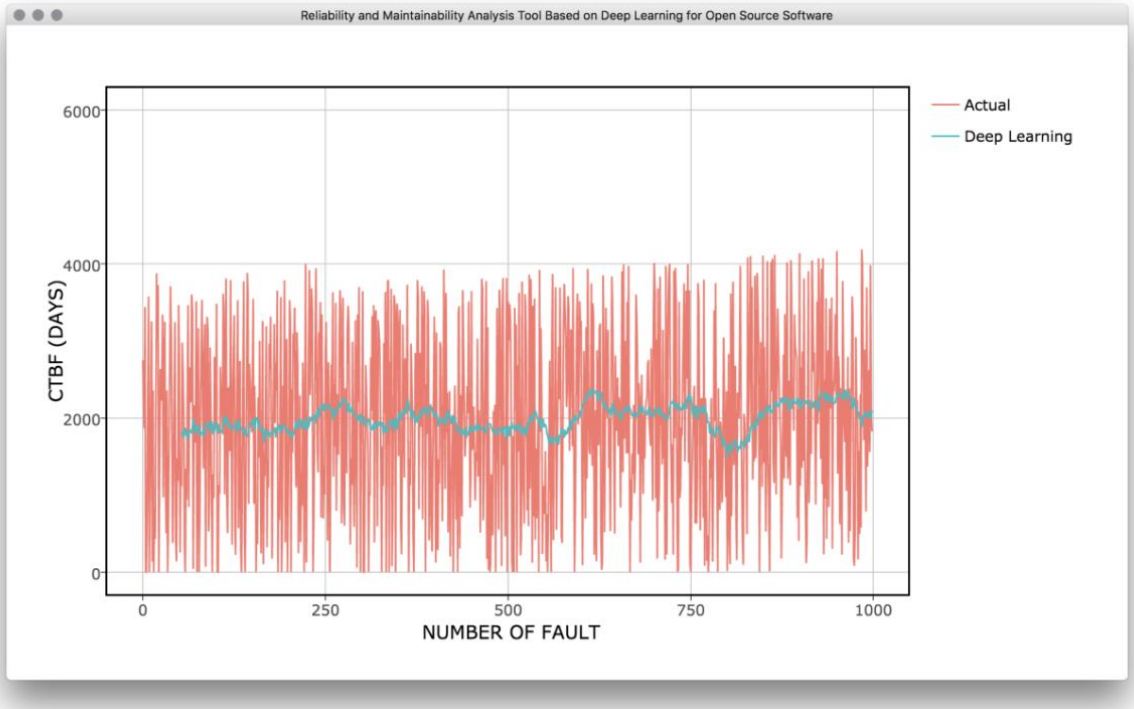

Fig. 7The estimated 50-fault moving-average CTBF based on deep learning.

IV. CONCLUDING REMARKS

At present, many OSS are developed under several OSS projects. Then, the bug tracking system is well known as the OSS development support tool based on the fault database. In particular, many types of software fault are recorded on the bug tracking systems.It will be useful for the software managers to understand the OSS reliability and maintainability,if many data sets recorded on the bug tracking systems are used for OSS project management.

This paper has focused on the method of OSS reliability and maintainability assessment based on the deep learning by using fault big data on bug tracking system. We have developed the method of reliability and maintainability assessmentbased on deep learning. In particular, the developed application software based on deep learning is used the latest technologies such as NW.js, statistical computing R, HTML, JavaScript, CSS3, ggplot2, and plotly. Then, we can easily implement as the application softwareby using NW.jsand plotly library of R. Also, several numerical illustrations of OSS reliability and maintainability assessment by using the fault data in the actual OSS project have been shown in this paper. Moreover, we have shown the performance illustrations of ourapplication software. Thereby, we have found that our tool can assist OSS reliability and maintainability based on the data on bug tracking system. The developed application software is available from "**http://www.comm.tcu.ac.jp/~tamura/**".

ACKNOWLEDGMENT

This work was supported in part by the JSPS KAKENHI Grant No. 15K00102 and No. 16K01242 in Japan.

REFERENCES

[1] M.R. Lyu, ed., *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, Los Alamitos, CA, 1996.

[2] S. Yamada, *Software Reliability Modeling: Fundamentals and Applications*, Springer-Verlag, Tokyo/Heidelberg, 2014.

[3] P.K. Kapur, H. Pham, A. Gupta, and P.C. Jha, *Software Reliability Assessment with OR Applications*, Springer-Verlag, London, 2011.

[4] S. Yamada and Y. Tamura, *OSS Reliability Measurement and Assessment*, Springer-Verlag, Switzerland, 2016.

[5] D.P. Kingma, D.J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models,"*Proceedings of Neural Information Processing Systems*, 2014.

[6] A. Blum, J. Lafferty, M.R. Rwebangira, and R. Reddy, "Semi-supervised learning using randomized mincuts,"*Proceedings of the International Conference on Machine Learning*, 2004.

[7] E.D. George, Y. Dong, D. Li, and A. Alex, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,"*IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp.30-42, 2012.

[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol, "Stacked denoisingautoencoders: Learning useful representations in a deep network with a local denoising criterion,"*Journal of Machine Learning Research*, Vol. 11, No. 2, 3371-3408, 2010.

[9] H.P. Martinez, Y. Bengio, and G.N. Yannakakis, "Learning deep physiological models of affect,"*IEEE Computational Intelligence Magazine*, Vol. 8, No. 2, pp. 20-33, 2013.

[10] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks,"*IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1944-1957, 2013.

[11] NW.js community, NW.js, http://nwjs.io/

[12] The R Project for Statistical Computing, The R Foundation, https://www.r-project.org/

[13] Plotly R Library, Plotly,https://plot.ly/r/

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