2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)

Performance Comparison of Bayesian Estimations on the Residual Number of Software Bugs

Yuki Hagihara[1], Tadashi Dohi[1,*], and Hiroyuki Okamura[1]
[1]1-4-1 Kagamiyama, Higashi-hiroshima, Hiroshima, Japan
d220580@hiroshima-u.ac.jp, dohi@hiroshima-u.ac.jp, okamu@hiroshima-u.ac.jp
*corresponding author

***Abstract*—In this paper, we consider the posterior distributions of the residual number of software bugs in a heterogeneous testing environment with different bug detection probabilities on each testing day, and compare two prior distributions; Poisson prior and negative binomial prior, where the former corresponds to the common non-homogeneous Poisson process (NHPP)-based software reliability model (SRM), the latter to the non-homogeneous mixed Poisson process (NHMPP)-based SRM. Throughout numerical experiments with an actual software bug count data, it is shown that the common NHPP-based SRM with the Poisson prior outperformed the NHMPP-based SRM with the negative binomial prior in terms of the accurate prediction of the residual number of software bugs.**

Index Terms*—*software reliability; Bayesian estimation; residual number of software bugs; discrete-time model; Gibbs sampler; Markov chain Monte Carlo; WAIC.

1. Introduction

Software plays a central role to control computer-based systems and influences the system dependability. Hence, software reliability is one of the most crucial dependability attributes to be guaranteed before releasing to the user or market. In predicting the residual number of software bugs on software programs, it is important to obtain the predictive distribution or the posterior distribution of the residual number of bugs by means of the Bayesian approach. Since the seminal contributions by Jewell [1],

Mazzuchi and Soyer [2], Csenki [3], many authors concerned the Bayesian software reliability modeling in the literature. An excellent survey on the Bayesian software reliability approach was presented in Singpurwalla and Wilson [4], [5]. A comprehensive approach via the Markov chain Monte Carlo (MCMC) was introduced to a homegeneouss Markov chain-based software reliability model (SRM) and the non-homogeneous Poisson process (NHPP)-based SRMs by Kuo and Yang [6], [7], respectively. Their computational approach is quite plausible from the practical points of view. Ramirez Cid and Achcar [8] also dealt with a specific NHPP-based SRM with nonmonotonic intensity functions by the MCMC approach.

When the software bug count is obsereved as group data which consist of the number of software bugs detected in each testing interval, it is appropriate to consider a discrete-time model for predicting the residual number of software bugs in the remaining testing period. Rallis and Lansdowne [9] considered a discrete-time SRM with the Poisson prior, which is essentially same as an NHPP-based SRM in continuous time, and investigated the posterior distribution of the residual number of software bugs in the remaining testing period. Chun [10] also treated a discrete-time model with the negative binomial prior in a different context of sequential inspection planning, and considered the similar problem to Rallis and Lansdowne [9] in a homogeneous testing environment with equal bug-detection probability over a planning horizon. The discrete-time model by Chun [10] corresponds to a homogeneous mixed Poisson process (HMPP) in continuous time. Okamura and Dohi [11] examined the non-homogeneous mixed Poisson process (NHMPP)-based SRMs with time-varying bug-detection probability, and showed that the goodness-of-fit performances of the NHMPP-based SRMs are exactly same as ones of the NHPP-based SRMs. Recently, Li et al. [12] generalized both NHPP- and NHMPP-based SRMs as non-homogeneous Markov processes and proposed a unified modeling approach.

In this paper we generalize the discrete-time model with equal bug-detection probabilities in Chun [10] by assuming different bug-detection probabilities over testing time, and focus on the residual number of software bugs. More specifically, we compare our negative binomial case with the Poisson prior case by Rallis and Lansdowne [9] in terms of the predictive performance of the residual number of software bugs, and investigate the posterior distributions of the residual number of software bugs in a heterogeneous testing environment with different bug detection probabilities on each testing day. In our discrete-time model, we assume 5 kinds of bug-detection probabilities to describe homogeneous and heterogeneous testing environments, and develop the Bayesian estimation, where two cases with the Poisson and the negative binomial priors are examined for the residual number of software bugs. In these cases, the former corresponds to the common non-homogeneous Poisson process (NHPP)-based SRM, the latter to the non-homogeneous mixed Poisson process (NHMPP) [11].

Especially we derive analytically the posterior distribution of the residual number of software bugs in the negative binomial prior case. Next, we consider more general cases in which the hyperparameters in bug-detection probabilities are also unknown, and apply the Gibbs sampling algorithms [6], [7] by means of the standard Markov chain Monte Carlo (MCMC). For our 2×5 discrete-time SRMs in the Bayesian estimation framework, it is worth mentioning that the com-

2325-6664/24/$31.00 ©2024 IEEE
DOI 10.1109/DSN-W60302.2024.00015

mon Akaike information criterion (AIC) [13] and Bayesian information criterion (BIC) [14] are not available, because the underlying parameter estimation method is not the maximum likelihood method. In this paper we apply the WAIC (widely applicable information criterion) by Watanabe [15] for the model selection of our Bayesian discrete-time SRMs. To our best knowledge, this is the first paper to deal with the model selection of Bayesian SRMs.

The paper is organized as follows. In Section 2, we summarize two discrete-time SRMs with Poisson and negative binomial priors in a heterogeneous testing environment with different bug detection probabilities. Section 3 concerns the Bayesian estimation for the residual number of software bugs. In Section 4, we discuss the model selection with the WAIC. Section 5 is devoted to numerical experiment with an actual software bug count data. Here we turn the hyperparameters in our Bayesian SRMs and select the best goodness-of-fit model with the WAIC [15]. The posterior distributions of the residual number of software bugs are compared in terms of the predictive performance. It is shown that the common NHPP-based SRM with the Poisson prior outperformed the NHMPP-based SRM with the negative binomial prior in terms of the accurate prediction of the residual number of software bugs. Finally, the paper is concluded with some remarks in Section 6.

2. Software Reliability Modeling

2.1. Model Description

Consider the software bug-detection process in discrete time measured by the calender day or week. Let x_i be the number of software bugs detected at the i-th $(i = 1, 2, \ldots)$ testing time, where $s_i = \sum_{j=1}^{i} x_j$ is the cumulative number of software bugs detected up to the i-th testing time. Suppose that each software testing is statistically independent. Once one software bug was detected, it is fixed or removed immediately, and no new software bug is introduced in the debugging. We assume the heterogeneous software testing circumstance with probabilities p_i $(i = 1, 2, \ldots)$ to detect each software bug at the i-th testing, where $q_i = 1 - p_i$. Let N and X_i be independent integer-valued random variables and denote the initial number of residual software bugs before the testing and the number of software bugs detected at the i-th testing, respectively.

From the independence assumption, the probability mass function (p.m.f.) of X_i conditioning that $N - s_{i-1}$ software bugs were remained by the i-th testing is given by the binomial p.m.f.:

$$P(X_i = x_i | N - s_{i-1}, p_i) = \binom{N - s_{i-1}}{x_i} p_i^{x_i} q_i^{N - s_i - x_i}. \quad (1)$$

Let $\boldsymbol{X} = \{X_1, X_2, \ldots, X_k\}$ and $\boldsymbol{p} = \{p_1, p_2, \ldots, p_k\}$ denote the random vector of X_i and the probability vector of p_i, respectively, by the k-th testing. Also, let $\boldsymbol{x} = \{x_1, x_2, \ldots, x_k\}$ be the realizations of the random vector $\boldsymbol{X}$. Then the likelihood function is equivalent to the following joint p.m.f.:

$$P(\boldsymbol{X} = \boldsymbol{x} | N, \boldsymbol{p}) = \frac{N!}{(N - s_k)! \sum_{i=i}^{k} x_i!} \sum_{i=1}^{k} p_i^{x_i} q_i^{N - s_i}. \quad (2)$$

2.2. Software Bug Detection Probabilities

In our problem, the probability vector $\boldsymbol{p}$ and their hyperparameters are unknown. Following the reference [16], we introduce 5 software bug detection probabilities. First of all, we consider the homogeneous testing in model0, where

$$p_i = \mu \quad (0 < \mu < 1). \quad (3)$$

Next we describe the heterogeneous testing environment with model1 $\sim$ model4. In model1, we suppose that the software bug detection probability p_i is given by

$$p_i = 1 - \frac{\mu}{\theta i + 1} \quad (0 < \mu < 1, \theta > 0), \quad (4)$$

which is due to Padgett and Spurrier [17]. In model2, the software bug detection probability is represented by the discrete log-logistic hazard rate [18];

$$p_i = \frac{1 - \mu}{\mu^{\ln i - \gamma + 1} + 1} \quad (0 < \mu < 1, -\infty < \gamma < \infty). \quad (5)$$

The model3 is characterized by the discrete pareto hazard rate [19];

$$p_i = 1 - \mu^{\ln(i+2)/(i+1)} \quad (0 < \mu < 1). \quad (6)$$

Finally, model4 is given by the well-known discrete Weibull hazard rate [20];

$$p_i = 1 - \mu^{i^\omega - (i-1)^\omega} \quad (0 < \mu < 1, 0 < \omega < 1). \quad (7)$$

For the relationship between these bug detection probabilities and software reliability modeling, see Zhao et al. [16]. In the latter discussion, we use the hyperparameter vector $\boldsymbol{\zeta}$ in the bug detection probability p_i for notational convenience, such as $\boldsymbol{\zeta} = \mu, (\mu, \theta), (\mu, \gamma), (\mu, \omega)$.

3. Bayesian Approach

In the standard approach via the Bayesian method, the initial number of residual software bugs at time $i = 0$ is given by a random variable with the prior distribution $P(N = n)$. From the well-known Bayes' theorem, the posterior distribution of N is represented with the likelihood function $P(\boldsymbol{X}|N, \boldsymbol{p})$ by

$$P(N = n | \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{p}) = \frac{P(N = n) P(\boldsymbol{X} = \boldsymbol{x} | n, \boldsymbol{p})}{\int P(N = n) P(\boldsymbol{X} = \boldsymbol{x} | n, \boldsymbol{p}) dn}, \quad (8)$$

where the denominator involves the Stieltjes integrals with respect to n for a given $\boldsymbol{p}$. The commonly used technique to compute the posterior distribution in the Bayesian analysis is the the Markov Chain Monte Carlo (MCMC) method. Next we consider two prior distributions of N and characterize the initial number of software bugs for known probability vector $\boldsymbol{p}$.

3.1. Poisson Prior

Rallis and Lansdowne [9] assumed the Poisson prior distribution with parameter λ_0 (> 0) for the initial softwae bug contents N. For given the probability vector $\boldsymbol{p}$, the following result can be easily obtained from the Bayes' theorem [9].

Proposition 1 [9]: The posterior distribution of the residual number of software bugs at the k-th testing is given by

$$P(N = n|\boldsymbol{x}, \boldsymbol{p}, \lambda_0) = \frac{\lambda_k^n}{n!} \exp\{-\lambda_k\} \tag{9}$$

with

$$\lambda_k = \lambda_0 \prod_{i=1}^{k} q_i. \tag{10}$$

3.2. Negative Binomial Prior

We consider the case where the initial number of bugs N has the negative binomial distribution with parameter α_0 (> 0) and $\beta_0 \in (0, 1)$. Chun [10] derived the posterior distribution of N when the software testing is homogeneous, say, $p_i = p$ $(i = 1, 2, \ldots)$. The following result is an extension of Chun [10] in a heterogeneous testing environment.

Proposition 2: The posterior distribution of the residual number of software bugs at the k-th testing is given by

$$\begin{aligned} P(N = n|\boldsymbol{x}, \boldsymbol{p}, \alpha_0, \beta_0) \quad = \quad & \binom{N + \alpha_k - 1}{n} (\beta_k)^{\alpha_k} \\ & \times (1 - \beta_k)^n, \end{aligned} \tag{11}$$

where

$$\alpha_k = \alpha_0 + s_k, \tag{12}$$

$$\beta_k = \beta_0 \prod_{i=1}^{k} q_i. \tag{13}$$

The proof is made by the inductive argument. Note that when $p_i = p$, i.e., in a homogeneous testing environment, Eq.(11) is reduced to the result in [10].

3.3. Gibbs Algorithms

The MCMC is used to approximate the posterior distributions, $P(N = n|\boldsymbol{x}, \lambda_0, \boldsymbol{p})$ and $P(N = n|\boldsymbol{x}, \alpha_0, \beta_0, \boldsymbol{p})$. For a parametric model $\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{\zeta})$, suppose that the hyperparameter, λ_0, α_0, β_0, $\boldsymbol{\zeta}$, are all uniformly distributed random variables. More precisely, it is assumed that λ_0 and α_0 obey the uniform distributions with support $(0, \lambda_{max})$ and $(0, \alpha_{max})$, respectively, and that β_0 has a support $(0, 1)$. For the software bug detection probabilities $\boldsymbol{p}(\boldsymbol{\zeta})$, the vector $\boldsymbol{\zeta}$ follows a uniformly distributed random vector. For instance, consider model1 $\boldsymbol{\zeta} = (\mu, \theta)$, where μ and θ are given by the independent and uniformly distributed random variables with support $(0, 1)$ and $(0, \theta_{max})$, respectively. Then, for given upper limits, λ_{max}, α_{max} and θ_{max}, the Gibbs algorithms in two cases with the Poisson and negative binomial priors are given by

Poisson Prior Case:

$$N \sim \text{Poisson}(\lambda_0) \tag{14}$$

$$\lambda_0 \sim \text{Uniform}(0, \lambda_{max}) \tag{15}$$

$$\mu \sim \text{Uniform}(0, 1) \tag{16}$$

$$\theta \sim \text{Uniform}(0, \theta_{max}). \tag{17}$$

Negative Binomial Prior Case:

$$N \sim \text{NegativeBinomial}(\alpha_0, \beta_0) \tag{18}$$

$$\alpha_0 \sim \text{Uniform}(0, \alpha_{max}) \tag{19}$$

$$\beta_0 \sim \text{Uniform}(0, 1) \tag{20}$$

$$\mu \sim \text{Uniform}(0, 1) \tag{21}$$

$$\theta \sim \text{Uniform}(0, \theta_{max}). \tag{22}$$

It is straightforward to derive the Gibbs algorithms for the other bug detection probabilities in Eqs.(3), (5), (6), (7). In the above Gibbs sampling schemes, we use the Gibbs sampling tool, called JAGS (just another Gibbs sampler) [21] by turning the hyperparameters, λ_{max}, α_{max} and θ_{max}. For the principle of MCMC with the Gibbs algorithms, see Kuo and Yang [6], [7].

4. Model Selection

4.1. WAIC

The determination of the hyperparameters $(\lambda_{max}, \alpha_{max}, \theta_{max})$ is made by means of WAIC (widely applicable information criterion). Watanabe [15] proposed WAIC and generalized the well-known AIC (Akaike information criterion) to check the goodness-of-fit for arbitrary statistical models. For the random vector $\boldsymbol{X} = \{X_1, X_2, \ldots, X_k\}$, WAIC is defined by

$$\text{WAIC} = T_k + \frac{V_k}{k}, \tag{23}$$

where

$$T_k = -\frac{1}{k} \sum_{i=1}^{n} \log p^*(X_i) \tag{24}$$

and

$$V_k = \sum_{i=1}^{k} \{\mathrm{E}_{\boldsymbol{\omega}}[(\log p(X_i|\boldsymbol{\omega}))^2] - (\mathrm{E}_{\boldsymbol{\omega}}[\log p(X_i|\boldsymbol{\omega})])^2\}. \tag{25}$$

Let $p(x|\boldsymbol{\omega})$ be the probability distribution of X with unknown parameter having the prior distribution. If the predictive distribution of X, $p^*(x)$, exists, the function T_k in Eq.(24) is called the *learning loss*. Letting $\mathrm{E}_{\boldsymbol{\omega}}$ denote the expectation with respect to $\boldsymbol{\omega}$, V_k in Eq.(25) is called the *functional variance*. So, WAIC is represented by the sum of learning loss and the functional variance. The smaller WAIC the better goodness-of-fit. Note again that the AIC can be validated for only the maximum likelihood estimation. On one hand, the WAIC enables us to determine the hyperparameters in the Bayesian estimation. We also determine λ_{max}, α_{max} and θ_{max} by minimizing WAIC in our analysis.

4.2. Convergence Diagnostics

In the MCMC, it is necessary to check whether the posterior distribution obtained by sampling truly converges to a stationary distribution. We apply two convergence diagnostics techniques; *Gelman-Rubin convergence diagnostics* [22] and *Geweke convergence diagnostics* [23], which are implemented in JAGS [21].

Gelman and Rubin [22] calculated the potential scale reduction factor (PSRF) from two or more chains of MCMC, and showed that the MCMC experiences sufficiently long computation period until the convergence to the steady state if PSRF is less than 1.1, where

$$\mathrm{PSRF} = \sqrt{\frac{\hat{V}}{W}} \tag{26}$$

with

$$W = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{X_{ij}-\bar{X}_i}{m(n-1)}, \tag{27}$$

$$\hat{V} = \frac{n-1}{n}W + \frac{B}{n}, \tag{28}$$

$$\frac{B}{n} = \sum_{i=1}^{m}\frac{(\bar{X}_i-\bar{X})^2}{m-1}. \tag{29}$$

In the above expressions, it is assumed that the MCMC generates m (≥ 2) chains and samples n times, so we have $X_{i1}, X_{i2}, \ldots, X_{in}$ $(i = 1, 2, \ldots, m)$, where $\bar{X}_i$ is the sample mean for a fixed i. Also, W in Eq.(27) and $\hat{V}$ in Eq.(28) are called the *within-chain variance estimate* and the *pooled variance estimate*, respectively.

Geweke [23] proposed another convergence diagnostics called the Geweke statistic or the Geweke convergence diagnostics by

$$Z = \frac{\bar{g}_{n_A} - \bar{g}_{n_B}}{V(\bar{g}_{n_A}) - V(\bar{g}_{n_B})}, \tag{30}$$

where $\bar{g}_{n_A}$ and $\bar{g}_{n_b}$ are the sample means of the first n_A data and the last n_B data, respectively, after n iterations of the Markov chain, and $V(\cdot)$ in Eq.(30) denotes their variances. If the absolute value of Z is less than 1.96, we judge that the MCMC converges to the steady state. In this paper we apply the above convergence diagnostics criteria to check the MCMC convergence.

5. Numerical Experiment

5.1. Experimental Setup

Our concern in this paper is to compute the posterior distributions of the residual number of software bugs after the testing with the Poisson and negative binomial prior cases in our discrete modeling. Figure 1 depicts the software bug count data used in the analysis [24], where 136 software bugs are found during 96 testing days in a real time command and control system.

To investigate the predictive performance of our discrete-time Bayesian SRMs, we obtain the posterior distributions of the residual number of software bugs at the observation points;

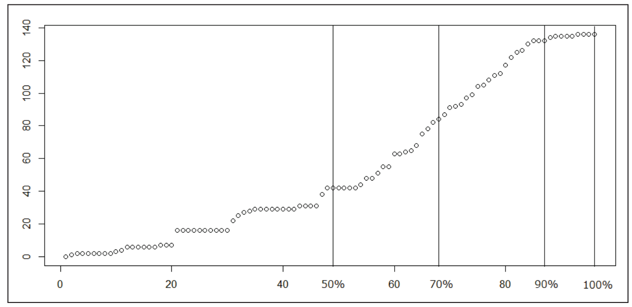


Fig. 1: Dataset.

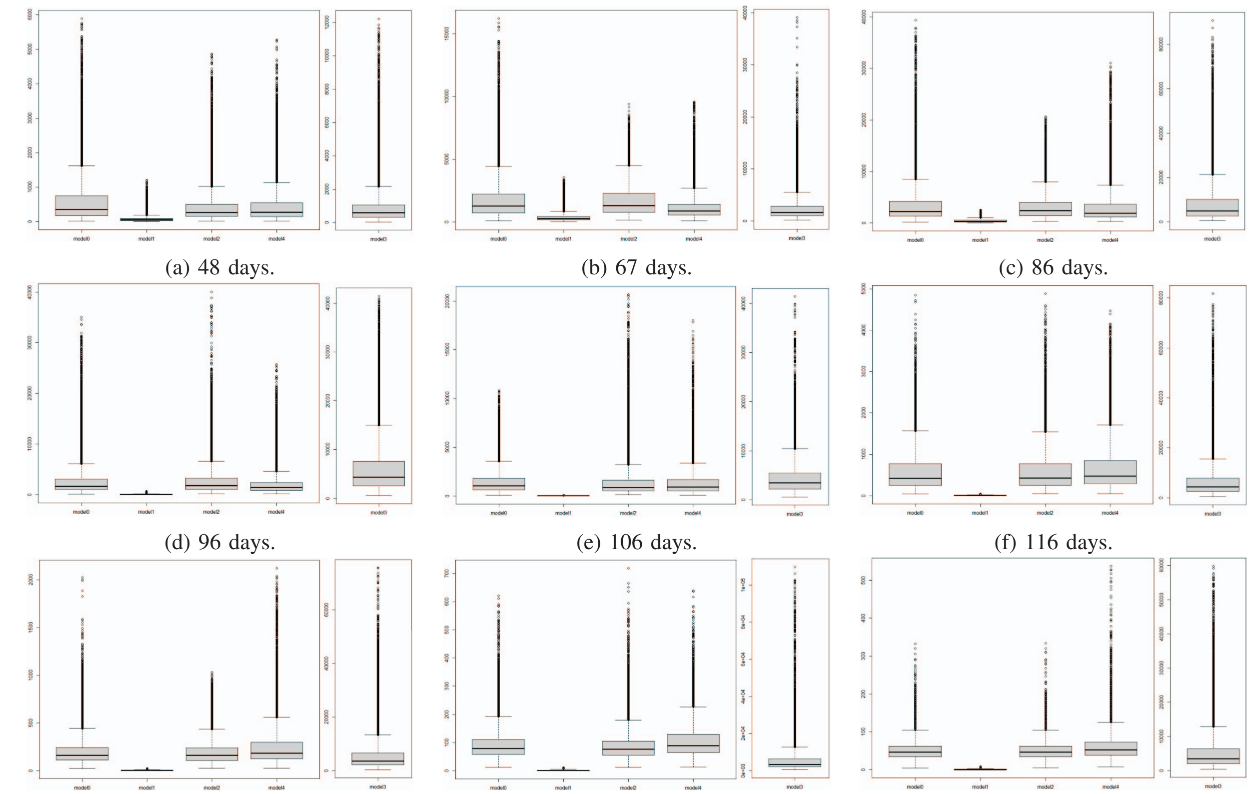
50%, 70%, 90% and 100% of the whole data. When predicting the initial number of residual software bugs in the early testing phase, it is known that the result tends to overestimate the actual software bug counts. Here, we consider the virtual testing [16] under the hypothesis that no software bug will be found after releasing the software at each observation point, and that the posterior distribution is estimated based on zero-count software bugs just after the observation point.

In our experiments, we apply 5 kinds of bug detection probabilities; model1 $\sim$ model4 in Eqs.(3)-(7) in two cases with the Poisson and negative binomial priors, where the hyperparameters (upper limits of the uniform distributions), $\lambda_{max}, \theta_{max}, \alpha_{max}$, are determined so as to minimize WAIC in the Gibbs algorithms in Eqs.(14)-(22). Once the parameters are estimated, we derive the posterior distribution of the residual number of software bugs at each observation point. In the data set of Fig. 1, 136 software bugs were detected at 100% observation point and zero-count period starts after 100% observation point (96th testing day). We continuously observe the zero-count until the 106th, 116th, 126th, 136th and 146th testing days to predict the residual number of software bugs.

5.2. Posterior Distributions

Table I presents the comparison of WAIC in two cases with the Poisson and negative binomial priors for 5 bug detection probability models. It can be seen that model1 could always provide the smaller WAIC at all observation points. Hence we conclude that model1 (Padgett and Spurrier model [17]) in the heterogeneous testing environment was the best among 5 bug detection probabilities. In Figs. 2 and 3, we show the box plots of posterior distributions with the Poisson and negative binomial priors, respectively, at all the observation points. In Fig. 2, it is found that the posterior distribution with model1 has smaller mean and variance than the other models, when the Poisson prior is assumed. On one hand, in Fig. 3, it is seen that no remarkable difference between model1 and the other models was found when the negative binomial prior was assumed. Especially, when the observation points increase, the posterior distributions approach to the degenerate distribution having the mass at the origin.

It is worth noting that the initial number of software bugs in our dataset for a legacy system [24] is known as 136. Hence, we compare the predictive performances of two prior cases at



(a) 48 days. (b) 67 days. (c) 86 days.

(d) 96 days. (e) 106 days. (f) 116 days.

(g) 126 days. (h) 136 days. (i) 146 days.

Fig. 2: Box plots of posterior distributions with the Poisson prior.

TABLE I: Comparison of WAIC.

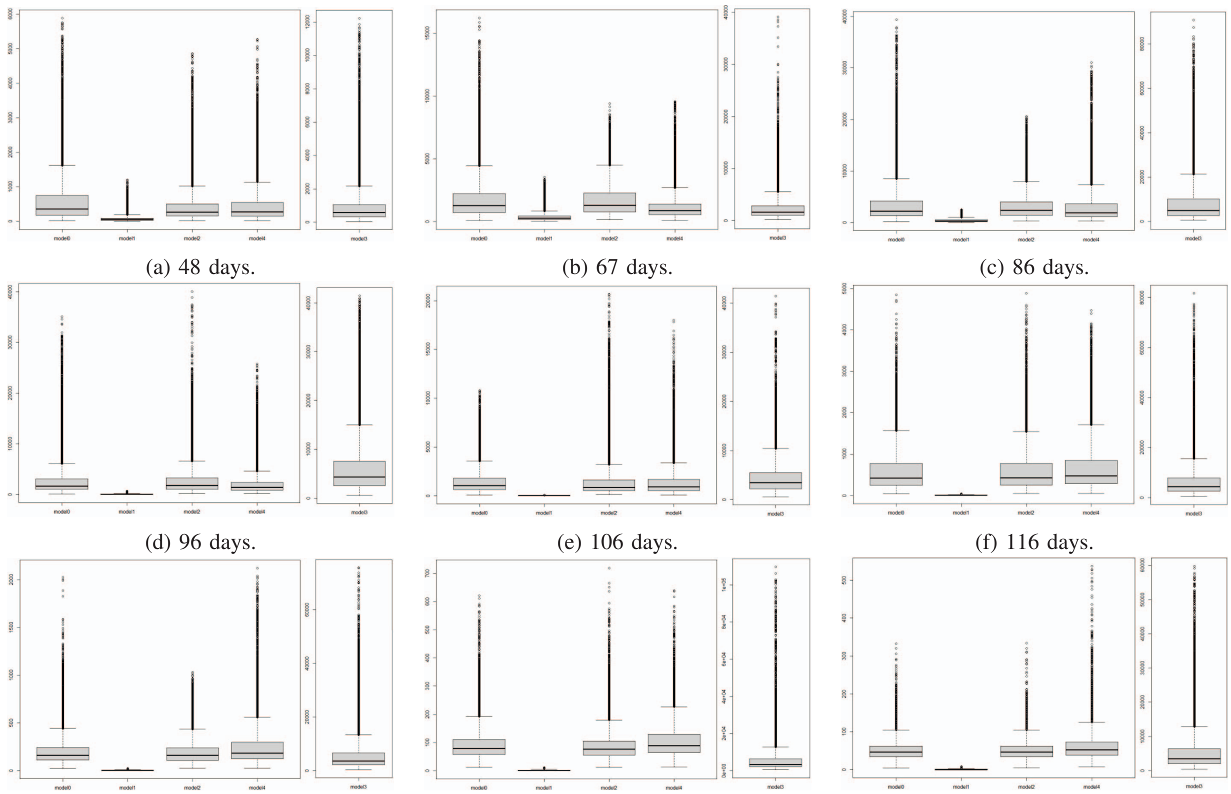
(i) Poisson prior.

	model0	model1	model2	model3	model4
48days	171.812	168.560	171.834	223.083	174.228
67days	279.330	255.040	279.272	413.673	281.482
86days	367.811	326.655	367.718	595.265	369.921
96days	393.398	364.345	393.598	611.735	395.727
106days	418.297	386.866	418.256	617.498	420.428
116days	440.910	396.445	441.068	622.616	442.720
126days	459.858	400.226	459.919	627.270	461.738
136days	473.638	401.338	473.601	631.419	475.548
146days	483.698	401.167	483.773	635.581	485.625

(ii) Negative binomial prior.

	model0	model1	model2	model3	model4
48days	172.171	169.015	172.500	223.320	174.865
67days	277.940	255.529	277.685	410.388	280.857
86days	363.364	327.085	363.031	583.474	365.684
96days	390.816	365.837	390.665	600.037	393.495
106days	417.468	387.096	417.704	606.790	419.795
116days	441.743	396.405	441.726	611.926	443.364
126days	461.576	400.232	461.417	617.602	463.340
136days	475.630	401.340	475.550	622.085	477.545
146days	485.402	401.234	485.153	626.537	487.965

each observation point. In Tables II, III, IV and V, we compare the means, medians, modes and standard deviations of the posterior distributions of the residual number of software bugs at each observation point, where the values in parentheses denote the differences from the actual number of software bugs, say, 136. Similar to the model selection based on WAIC, it is shown that model1 gave the much smaller prediction values in Tables II, III and IV. In these results, since no software bug was detected after 96th testing day, the number of residual software bugs is expected to be close to 0 in the heterogeneous testing. At the first look the Poisson and negative binomial prior cases, there are not so remarkable differences in the prediction. However, careful comparison suggests that the common NHPP-based SRM with the Poisson prior outperformed the NHMPP-based SRM with the negative binomial prior in terms of the accurate prediction of the residual number of software bugs from the viewpoint of the mean values in Table II. Looking at the medians in Table III, it is seen that both the Poisson and negative prior cases gave the exactly same prediction results, but the modes of the posterior distributions are rather different in Table IV. In Table V, we compare the standard deviations of the posterior distributions. It can be confirmed that model1 always gave smaller standard deviations of the posterior distributions. Even in comparison of the standard deviations, the Poisson prior provided smaller standard deviations than the negative binomial prior, so we conclude that the Poisson prior enabled us to derive more accurate prediction of the residual number of software bugs with less variability.



(a) 48 days. (b) 67 days. (c) 86 days.

(d) 96 days. (e) 106 days. (f) 116 days.

(g) 126 days. (h) 136 days. (i) 146 days.

Fig. 3: Box plots of the posterior distributions with the negative binomial prior.

6. Conclusions

In this paper, we have considered the posterior distributions of the residual number of software bugs in a heterogeneous testing environment with different bug detection probabilities on each testing day, and compared two prior distributions; Poisson prior and negative binomial prior. We have applied the standard Bayesian method to estimate the residual number of software bugs, where 5 bug detection probabilities were examined. By means of the familiar MCMC technique, we have predicted the residual number of software bugs and carried out the model selection with the WAIC. In numerical experiment with an actual software bug count data, we have shown that the common NHPP-based SRM with the Poisson prior outperformed the NHMPP-based SRM with the negative binomial prior in terms of the accurate prediction of the residual number of software bugs, and that the bug detection probability model by Padgett and Spurrier [17] was most approptiate to represent the heterogeneous software testing.

In this paper we have implicitly assumed that the hyperparameters had the non-informative uniform priors. As another non-informative priors, it is also interesting to apply the Jeffereys prior and to compare it with our results. Although we have treated only one dataset in this paper for brevity, the comparison between the Poissson and negative binomial priors should be made with more data sets. In future, we will conduct a comprehensive numerical experiment with multiple datasets and investigate the dependence of the prior distributions in software reliability modeling and software bug prediction.

References

[1] W. S. Jewell, Bayesian extensions to a basic model of software reliability, *IEEE Transactions on Software Engineering*, vol. 11, no. 12, pp. 1464–1471, 1985.

[2] T. A. Mazzuchi and R. Soyer, A Bayes empirical-Bayese model for software reliability, *IEEE Transactions on Reliability*, vol. 37, no. 2, pp. 248–254, 1988.

[3] A. Csenki, Bayes predictive analysis of a fundamental software reliability model, *IEEE Transactions on Reliability*, vol. 39, no. 2, pp. 177–183, 1990.

[4] N. D. Singpurwalla and S. P. Wilson, Software reliability modeling, *International Statistical Review*, vol. 62, no. 3, pp. 289–317, 1994.

[5] N. D. Singpurwalla and S. P. Wilson, *Statistical Methods in Software Engineering: Reliability and Risk*, Springer, 1999.

[6] L. Kuo and T. Yang, Bayesian computation of software reliability, *Journal of Computation and Graphical Statistics*, vol. 4, pp. 65-–72, 1995.

[7] L. Kuo and T. Yang, Bayesian computation of nonhomogeneous Poisson processes in software reliability, *Journal of the American Statistical Association*, vol. 91, pp. 763-–773, 1996.

[8] J. E. Ramirez Cid and J. A. Achcar, Bayesian inference for nonhomogeneous Poisson processes in software reliability models assuming nonmonotonic intensity functions, *Computation Statistics and Data Analysis*, vol. 32, pp. 147-–159, 1999.

TABLE II: Comparison of mean values of the posterior distributions.

(i) Poisson prior.

	model0		model1		model2		model3		model4	
48days	463.668	(+369.668)	99.550	(+5.550)	410.449	(+316.449)	567.337	(+473.337)	426.697	(+332.697)
67days	645.460	(+593.460)	302.626	(+250.626)	636.990	(+584.990)	718.583	(+666.583)	646.570	(+594.570)
87days	693.648	(+689.648)	372.634	(+368.634)	700.352	(+696.352)	771.733	(+767.733)	696.944	(+692.944)
96days	666.529	(+666.529)	49.641	(+49.641)	653.901	(+653.901)	769.374	(+769.374)	659.090	(+659.090)
106days	588.169	(+588.169)	21.685	(+21.685)	584.720	(+584.720)	760.982	(+760.982)	546.616	(+546.616)
116days	422.870	(+422.870)	8.108	(+8.108)	468.233	(+468.233)	758.775	(+758.775)	476.979	(+476.979)
126days	114.010	(+114.010)	3.438	(+3.438)	113.633	(+113.633)	759.641	(+759.641)	118.553	(+118.553)
136days	51.751	(+51.751)	1.517	(+1.517)	51.772	(+51.772)	751.457	(+751.457)	53.936	(+53.9356)
146days	41.264	(+41.264)	0.679	(+0.679)	41.212	(+41.212)	747.401	(+747.401)	43.798	(+43.798)

(ii) Negative binomial prior.

	model0		model1		model2		model3		model4	
48days	600.510	(+506.510)	80.789	(-13.211)	439.315	(+345.315)	939.875	(+845.875)	458.346	(+364.346)
67days	1812.550	(+1760.550)	370.585	(+318.585)	1634.923	(+1582.923)	2445.270	(+2393.270)	1159.827	(+1107.827)
87days	3513.810	(+3509.810)	442.222	(+438.222)	3296.680	(+3292.680)	8572.750	(+8568.750)	3096.590	(+3092.590)
96days	2931.090	(+2931.090)	87.286	(+87.286)	2899.800	(+2899.800)	6287.090	(+6287.090)	2166.050	(+2166.050)
106days	1412.623	(+1412.623)	22.174	(+22.174)	1522.650	(+1522.650)	4388.640	(+4388.640)	1435.928	(+1435.928)
116days	599.700	(+599.700)	8.088	(+8.088)	602.694	(+602.694)	6583.600	(+6583.600)	660.759	(+660.759)
126days	205.820	(+205.820)	3.454	(+3.454)	196.343	(+196.343)	6131.020	(+6131.020)	252.937	(+252.937)
136days	91.956	(+91.9564	1.532	(+1.532)	89.336	(+89.336)	5878.000	(+5878.000)	104.060	(+104.060)
146days	51.080	(+51.080)	0.690	(+0.690)	50.658	(+50.658)	5481.590	(+5481.590)	60.021	(+60.021)

TABLE III: Comparison of medians of the posterior distributions.

(i) Poisson prior.

	model0		model1		model2		model3		model4	
48days	435	(+341)	68	(-26)	377	(+283)	577	(+483)	392	(+298)
67days	670	(+618)	286	(+234)	655	(+603)	747	(+695)	667	(+615)
87days	716	(+712)	359	(+355)	727	(+723)	791	(+787)	718	(+714)
96days	687	(+687)	49	(+49)	677	(+677)	789	(+789)	688	(+688)
106days	610	(+610)	20	(+20)	600	(+600)	779	(+779)	560	(+560)
116days	411	(+411)	7	(+7)	450	(+450)	778	(+778)	458	(+458)
126days	114	(+114)	3	(+3)	114	(+114)	779	(+779)	120	(+120)
136days	51	(+51)	1	(+1)	51	(+51)	769	(+769)	54	(+54)
146days	40	(+40)	0	(+0)	40	(+40)	769	(+769)	43	(+43)

(ii) Negative binomial prior.

	model0		model1		model2		model3		model4	
48days	435	(+341)	68	(-26)	377	(+283)	577	(+483)	392	(+298)
67days	670	(+618)	286	(+234)	655	(+603)	747	(+695)	667	(+615)
87days	716	(+712)	359	(+355)	727	(+723)	791	(+787)	718	(+714)
96days	687	(+687)	49	(+49)	677	(+677)	789	(+789)	688	(+688)
106days	610	(+610)	20	(+20)	600	(+600)	779	(+779)	560	(+560)
116days	411	(+411)	7	(+7)	450	(+450)	778	(+778)	458	(+458)
126days	114	(+114)	3	(+3)	114	(+114)	779	(+779)	120	(+120)
136days	51	(+51)	1	(+1)	51	(+51)	769	(+769)	54	(+54)
146days	40	(+40)	0	(+0)	40	(+40)	769	(+769)	43	(+43)

[9] N. E. Rallis and Z. F. Lansdowne, Reliability estimation for a software system with sequential independent reviews, *IEEE Transactions on Software Engineering*, vol. 27, no. 12, pp. 105–1061, 2001.

[10] Y. H. Chun, Bayesian analysis of the sequential inspection plan via the Gibbs sampler, *Operations Research*, vol. 56, no.1, pp. 235–246, 2008.

[11] H. Okamura and T. Dohi, Software reliability modeling based on mixed Poisson distributions, *International Journal of Reliability, Quality and Safety Engineering,* vol. 15, no. 1, pp. 9–32, 2008.

[12] S. Li, T. Dohi and H. Okamura, Non-homogeneous Markov process modeling for software reliability assessment, *IEEE Transactions on Reliability*, vol. 72, no 4, pp. 1540–1555, 2023.

[13] H. Akaike, Information theory and an extension of the maximum likelihood principle, *Proceedings of The 2nd International Symposium on Information Theory*, B. N. Petrov and F. Caski (eds.), Akadimiai Kiado, pp. 267–281, Budapest, 1973.

[14] G. Schwar, Estimating the dimension of a model, *Annals of Statistics*, vol. 6, pp. 461—464, 1978.

[15] S. Watanabe, Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory, *Journal of Machine Learning Research*, vol. 11, pp. 3571–3594, 2010.

[16] Y. Zhao, T. Dohi and H. Okamura, Software test-run reliability modeling with non-homogeneous binomial processes, *Proceedings of The 23rd IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2018)*, pp.145–154, IEEE CPS, 2018.

[17] W. J. Padgett and J. D. Spurrier, On discrete failure models, *IEEE Transactions on Reliability*, vol. R-34, pp. 253—256, 1985.

[18] S. Chakraborty and D. Chakravarty, A new discrete probability distribution with integer support on $(-\infty, \infty)$, *Communications in Statistics - Theory and Methods*, vol. 45, no. 2, pp. 492—505, 2016.

[19] H. Krishna and P. S. Pundir, Discrete Burr and discrete pareto distributions, *Statistical Methodology*, vol. 6, no. 2, pp. 177-–188, 2009.

[20] T Nakagawa and S. Osaki, The discrete Weibull distribution, *IEEE Transactions on Reliability*, vol. R-24, pp. 300–301, 1975.

[21] M. Plummer, JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling, *Proceedings of The 3rd International Workshop on Distributed Statistical Computing*, 2003.

[22] A. Gelman and D. B. Rubin, Inference from iterative simulation using multiple sequences, *Statistical Science*, vol. 7, pp. 457-–472, 1992.

TABLE IV: Comparison of modes of the posterior distributions.

(i) Poisson prior.

	model0		model1		model2		model3		model4	
48days	183	(+89)	37	(-57)	128	(+34)	716	(+622)	229	(+135)
67days	796	(+744)	233	(+181)	811	(+759)	839	(+787)	809	(+757)
87days	779	(+775)	352	(+348)	815	(+811)	832	(+828)	794	(+790)
96days	755	(+755)	48	(+48)	694	(+694)	826	(+826)	795	(+795)
106days	724	(+724)	16	(+16)	781	(+781)	821	(+821)	690	(+690)
116days	369	(+369)	6	(+6)	274	(+274)	802	(+802)	319	(+319)
126days	118	(+118)	2	(+2)	11	(+11)	826	(+826)	123	(+123)
136days	53	(+53)	1	(+1)	51	(+51)	811	(+811)	55	(+55)
146days	38	(+38)	0	(+0)	38	(+38)	784	(+784)	39	(+39)

(ii) Negative binomial prior.

	model0		model1		model2		model3		model4	
48days	86	(-8)	22	(-72)	170	(+76)	293	(+199)	85	(-9)
67days	518	(+466)	157	(+105)	622	(+570)	1180	(+1128)	604	(+552)
87days	1123	(+1119)	246	(+242)	1144	(+1140)	2489	(+2485)	960	(+956)
96days	1359	(+1359)	50	(+50)	825	(+825)	3663	(+3663)	775	(+775)
106days	594	(+594)	17	(+17)	735	(+735)	2453	(+2453)	382	(+382)
116days	188	(+188)	7	(+7)	232	(+232)	2592	(+2592)	227	(+227)
126days	114	(+114)	2	(+2)	97	(+97)	2276	(+2276)	119	(+119)
136days	60	(+60)	1	(+1)	57	(+57)	2201	(+2201)	73	(+73)
146days	41	(+41)	0	(+0)	35	(+35)	1402	(+1402)	44	(+44)

TABLE V: Comparison of standard deviations of the posterior distributions.

(i) Poisson prior.

	model0	model1	model2	model3	model4
48days	261.698	90.337	224.094	234.229	242.717
67days	183.270	137.810	182.074	148.726	178.018
86days	131.137	150.850	132.898	90.687	129.684
96days	143.712	14.261	147.430	92.389	154.128
106days	179.241	9.843	181.830	91.858	146.422
116days	173.280	4.417	207.103	96.534	199.479
126days	32.465	2.390	32.322	95.133	32.064
136days	14.237	1.422	14.097	97.078	14.443
146days	13.685	0.894	13.855	102.824	14.103

(ii) Negative binomial prior.

	model0	model1	model2	model3	model4
48days	694.470	97.754	521.705	1208.249	539.719
67days	1758.070	351.784	1165.875	2517.790	1023.186
86days	3812.270	324.995	2724.210	10019.170	3365.420
96days	3635.740	52.815	3291.000	5902.760	2454.990
106days	1207.943	10.469	1983.410	3350.330	1588.454
116days	511.142	4.430	524.613	7002.390	556.864
126days	159.446	2.396	131.110	7416.680	218.850
136days	54.329	1.436	54.599	7820.080	59.175
146days	25.367	0.925	24.462	6094.790	35.061

[23] J. Geweke, Evaluating the accuracy of sampling-based approaches to calculating posterior moments, *Bayesian Statistics 4*, J. M. Bernado, J. O. Berger, A. P. Dawid and A. F. M. Smith (eds.), Clarendon Press, pp. 169-–193, Oxford, UK, 1992.

[24] J. D. Musa, *Software Reliability Data*, Technical Report in Rome Air Development Center, New Jersey, 1979.