

# Software Defects Prediction using Operating Characteristic Curves

Torsten Bergander\*    Yan Luo<sup>◇</sup>    A. Ben Hamza<sup>◇</sup>

\*SAP Labs Canada Inc., Montréal, QC, Canada

<sup>◇</sup>Concordia Institute for Information Systems Engineering  
Concordia University, Montréal, QC, Canada

## Abstract

We present a software defect prediction model using operating characteristic curves. The main idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast prediction method to accurately predict the cumulative number of failures at any given stage during the software development process. Our predictive approach uses the number of detected faults instead of the software failure-occurrence time in the testing phase. Experimental results illustrate the effectiveness and the much improved performance of the proposed method in comparison with the Bayesian prediction approaches.

## 1 Introduction

Each software defect encountered by customers entails a significant cost penalty for software companies. Thus, knowledge about how many defects to expect in a software product at any given stage during its development process is a very valuable asset. Being able to estimate the number of defects will substantially improve the decision processes about releasing a software product. Moreover, the production process for software products can be substantially improved by employing a prediction model that accounts for the dynamic nature of software production processes and reliably predicts the number of defects [1–5].

During the development process of computer software systems, many software defects may be introduced and often lead to critical problems and complicated breakdowns of computer systems [6]. Hence, there is an increasing demand for controlling the software development process in terms of quality and reliability. Software reliability can be evaluated by the number of detected faults. A software failure is defined as an unacceptable departure of program operation caused by a software fault remaining in the software system [1, 7]. In the traditional software development environment, software reliability evaluation, which shortens development intervals and reduces development costs, provides useful guidance in balancing reliability, time-to-market and development cost [4]. Hence, there is an increasing demand for prediction of the quality and reliability of software.

Several software reliability prediction models have been proposed in the literature for estimating system reliability,

but all these kinds of models make unrealistic assumptions to ensure solvability [7–14]. These unreasonable assumptions have limited the applications of these models [3, 5].

Bayesian statistics provide a framework for combining observed data with prior assumptions in order to model stochastic systems. Bayesian methods aim at assigning prior distributions to the parameters in the model in order to incorporate whatever *a priori* quantitative or qualitative knowledge we have available, and then to update these priors in the light of the data, yielding a posterior distribution via Bayes's Theorem [15]. The ability to include prior information in the model is not only an attractive pragmatic feature of the Bayesian approach, but it is also theoretically vital for guaranteeing coherent inferences.

Motivated by the widely used concept of operating characteristic (OC) curves in statistical quality control to select the sample size at the outset of an experiment [16], we propose in this paper a software defect prediction technique using OC curves in order to predict the cumulative number of failures at any given time. The core idea behind our proposed methodology is to use geometric insight in helping construct an efficient and fast prediction method to accurately predict the cumulative number of failures at any given time.

The layout of this paper is organized as follows. In the next Section, a problem formulation is stated. In Section 3, we briefly review some Bayesian prediction models that will be used for comparison with our proposed approach. In Section 4, we propose a new prediction algorithm based on OC curves. In Section 5, we present experimental results to demonstrate the much improved performance of the proposed approach in the prediction of software defects. Finally, some conclusions are included in Section 6.

## 2 Problem Formulation

Software failure data are usually available to the user in three basic forms:

1. in the form of a sequence of ordered failure times  
 $0 < t_1 < t_2 < \dots < t_n$
2. in the form of a sequence of interfailure times  $\tau_i$  where  
 $\tau_i = t_i - t_{i-1}$  for  $i = 1, \dots, n$
3. in the form of cumulative number of failures.

It is easy to verify that the failure and interfailure times are related by  $t_i = \sum_{j=1}^i \tau_j$ .

The cumulative number of failures  $N(t_i)$  detected by time  $t_i$  (i.e. the cumulative number of failures over the period  $[0, t_i)$ ) defines a non-homogeneous Poisson process (NHPP) with failure intensity or rate function  $\lambda(t_i)$  such that the rate function of the process is time-dependent. The mean value function  $m(t_i) = E(N(t_i))$  of the process is given by  $m(t_i) = \int_0^{t_i} \lambda(u)du$ . Moreover, the function

$$p(t_i) = \lambda(t_i) \exp\left(-\int_0^{t_i} \lambda(u)du\right) = \lambda(t_i) \exp(-m(t_i))$$

defines a probability density function.

On the other hand, the number of failures  $N(t_i, t_j)$  in any interval  $[t_i, t_j)$  defines a non-homogeneous Poisson process with mean function

$$\int_{t_i}^{t_j} \lambda(u)du = m(t_j) - m(t_i).$$

That is,

$$\begin{aligned} P(N(t_j) - N(t_i) = \kappa) \\ = \frac{(m(t_j) - m(t_i))^\kappa}{\kappa!} \exp(-(m(t_j) - m(t_i))). \end{aligned}$$

Software reliability  $R(t_j|t_i)$  is defined as the probability that no software failure is detected in the time interval  $(t_i, t_i+t_j)$ , given that the last failure occurred at testing time  $t_i$ , and it is given by

$$R(t_j|t_i) = \exp\left(-\left(m(t_i + t_j) - m(t_i)\right)\right).$$

It is worth pointing out that if the failure intensity function is time-independent, then the cumulative number of failures  $N(t_i)$  defines a homogeneous Poisson process (HPP).

Note that the interfailure times may have non-exponential distributions, and hence the cumulative number of failures  $N(t_i)$  would define a general renewal process.

The problem addressed in this paper may now be concisely described as follows: Given the historical failure times data  $\mathcal{D} = \{t_1, \dots, t_n\}$  and its corresponding cumulative number of failures data  $\mathcal{N} = \{N(t_1), \dots, N(t_n)\}$ , find the predicted cumulative number of failures at any given time  $t$ .

### 3 Prediction using Bayesian Statistics

Assume we model the failure times using an NHPP with a parametrized failure intensity function  $\lambda(t; \boldsymbol{\theta})$ , where  $\boldsymbol{\theta}$  is a vector of unknown parameters.

Consider the problem of making prediction for a new failure time  $t$  without any measurements on the predictors for any of the individuals so that the dataset is just given by  $\mathcal{D} = \{t_1, \dots, t_n\}$ . That is, we want to determine  $p(t|\mathcal{D})$ , the probability density function of the new failure time conditioned on the observed failure times. The function  $p(t|\mathcal{D})$  is referred to as *predictive density* of a new failure time and may be written in integral form as

$$p(t|\mathcal{D}) = \int p(t|\mathcal{D}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta},$$

where  $p(\boldsymbol{\theta}|\mathcal{D})$  is the posterior distribution of  $\boldsymbol{\theta}$  given by

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{\{\prod_{i=1}^n p(t_i|\boldsymbol{\theta})\}p(\boldsymbol{\theta})}{\int \{\prod_{i=1}^n p(t_i|\boldsymbol{\theta})\}p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

and  $p(\boldsymbol{\theta})$  is the prior distribution which represents information available about the unknown parameters. The prior estimate provides a means of combining exogenous information with observed data in order to estimate parameters of a probability distribution. It is convenient to choose simple forms of prior distributions which result in computationally tractable posterior distributions. Hence, the posterior distribution is found by combining the prior distribution  $p(\boldsymbol{\theta})$  with the probability  $p(\mathcal{D}|\boldsymbol{\theta})$  of observing the data given the parameters. The probability  $p(\mathcal{D}|\boldsymbol{\theta})$  is also called the likelihood function of the data and it is given by

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{i=1}^n p(t_i|\boldsymbol{\theta}),$$

where

$$p(t_i|\boldsymbol{\theta}) = \lambda(t_i; \boldsymbol{\theta}) \exp\left(-\int_0^{t_i} \lambda(u; \boldsymbol{\theta})du\right)$$

assuming that the failure times data are independent and identically distributed (iid). The likelihood function is the probability of observing the given data as a function of  $\boldsymbol{\theta}$ .

Hence, the Bayesian approach consists of three main steps:

1. Assign prior distributions to all the unknown parameters.
2. Determine the likelihood of the data given the parameters.
3. Determine the posterior distribution of the parameters given the data.

#### 3.1 Bayesian prediction

The Bayesian prediction approach proposed in [2] is based on the power law model shown in Table ???. The parameter  $b$  of the power law model may be estimated as follows

$$\hat{b} = \frac{t_n}{\sum_{t=t_1}^{t_n} \log[N(t_n)/N(t)]},$$

and the predicted cumulative number of defects  $N(t)$  at time  $t$  is given by

$$N(t) = N(t_n) \left(\frac{t}{t_n} F(2t, 2t_n; \gamma)\right)^{1/\hat{b}}, \quad (1)$$

where  $\gamma = P\{\chi_n^2 \leq \chi_{\gamma, n}^2\}$ , and  $F(2t, 2t_n; \gamma)$  denotes the  $\gamma$  percentage point of the  $F$ -distribution with  $2t$  and  $2t_n$  degrees of freedom.

#### 3.2 Bayesian prediction using MCMC

If we draw samples  $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$  from the posterior distribution  $p(\boldsymbol{\theta}|\mathcal{D})$ , then the predictive density may be approximated as follows

$$p(t|\mathcal{D}) \approx \sum_{i=1}^N p(t|\mathcal{D}, \boldsymbol{\theta}^{(i)})p(\boldsymbol{\theta}^{(i)}|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N p(t|\mathcal{D}, \boldsymbol{\theta}^{(i)}).$$

The samples  $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(N)}$  are draws from the posterior distribution of  $\boldsymbol{\theta}$ , and may be obtained using Markov chain Monte Carlo (MCMC) simulation algorithms [17, 18].

For the Bayesian prediction approach using MCMC, the predicted cumulative number of defects  $N(t)$  at time  $t$  is also given by Eq. (1) where  $\hat{b}$  is estimated using the MCMC algorithm [18].

#### 4 Proposed Method

Consider the two-sided hypothesis

$$\begin{aligned} H_0 &: t = t_k \\ H_1 &: t \neq t_k \end{aligned}$$

where  $H_0$  and  $H_1$  are the null and the alternative hypotheses respectively.

Define  $\chi_{\alpha,k}^2$  as the percentage value of the chi-square distribution with  $k$  degrees of freedom such that the probability that the chi-square distribution  $\chi_n^2$  exceeds this value is  $\alpha$ , that is

$$P\{\chi_k^2 \geq \chi_{\alpha,k}^2\} = \alpha = P\{\text{reject } H_0 | H_0 \text{ is true}\},$$

where  $\alpha \in (0, 1)$  is the probability of type I error (also referred to as the significance level).

Suppose that  $H_0$  is false and that the true value is  $t = t_k + \delta$ , where  $\delta > 0$ . Since  $H_1$  is true, the distribution of the test statistic

$$Z = \frac{\chi_t^2 - t_k}{\sqrt{2k}}$$

has a mean value equal to  $\delta/\sqrt{2k}$ , and a type II error will be made only if  $-\chi_{\alpha/2}^2 \leq Z \leq \chi_{\alpha/2}^2$ . That is, the probability of type II error  $\beta = P\{\text{accept } H_0 | H_0 \text{ is false}\}$  may be expressed as

$$\beta = \Phi\left(\chi_{\frac{\alpha}{2}, t}^2 - \frac{\delta}{\sqrt{2k}}\right) - \Phi\left(-\chi_{\frac{\alpha}{2}, t}^2 - \frac{\delta}{\sqrt{2k}}\right),$$

where  $\Phi$  is the cumulative distribution function of  $\chi_t^2$ .

The function  $\beta(t)$  is evaluated by finding the probability that the test statistic  $Z$  falls in the acceptance region given a particular value of  $t$ . We define the operating characteristic (OC) curve of a test as the plot of  $\beta(t)$  against  $t$ . Note that given the OC curve parameters  $\beta, \alpha, k$ , and  $\delta$ , we can derive the predicted cumulative number of defects at time  $t$  as follows

$$N(t) = \left(\frac{\sqrt{2k}}{\delta}\right)^2 \left(\chi_{\alpha,\delta}^2 + \chi_{\beta,\delta}^2\right). \quad (2)$$

Fig. 1 depicts a plot of the cumulative number of defects using OC curves.

The OC curve approach, however, makes a prediction without taking into account the historical data. To circumvent this limitation, we propose a predictive operating characteristic (POC) curve where the predicted cumulative number of defects at time  $t$  is calculated as follows

$$N(t) = \left(\frac{\sqrt{2p}}{\delta}\right)^2 \left(\chi_{\alpha,\delta}^2 + \chi_{\beta,\delta}^2\right), \quad (3)$$

and the parameter  $p$  is given by (see Fig. 2)

$$p = \begin{cases} N(t), & \text{if } t \leq t_n \\ N(t_n), & \text{if } t_n < t \leq T. \end{cases}$$

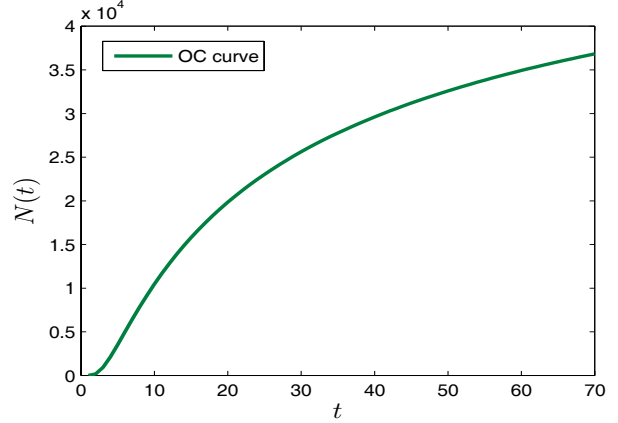


Fig. 1. Illustration of cumulative number of defects using OC curves.

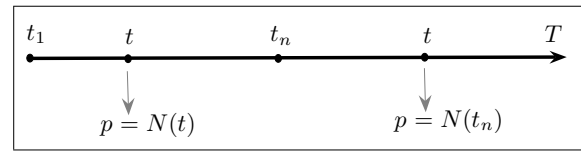


Fig. 2. Illustration of the  $p$  parameter in the POC curve.

#### 5 Experimental Results

We tested our proposed method on a real software failure dataset (DS I) that was taken from a SAP development system. This dataset contains monthly software failures that were recorded for a period of 60 months as shown in Table I.

Fig. 3 depicts the cumulative number of failures versus failure time (month) during a software life cycle.

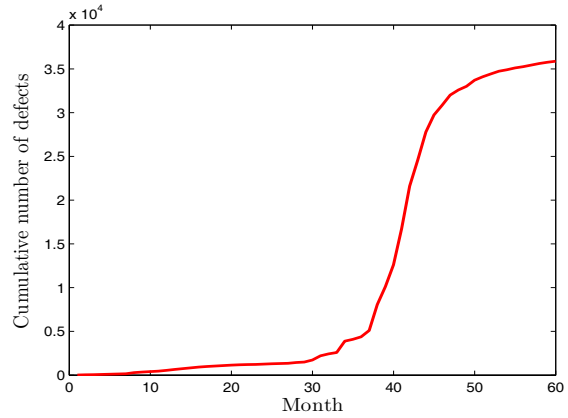


Fig. 3. Cumulative Number of Failures vs. Failure Time (DS I)

We also applied the proposed method to a truncated dataset (DS II) that was obtained by truncating the original software failure data after the 40th month as shown in Fig. 4. Note that the cumulative number of failures stabilizes substantially after the 50th month, which clearly indicate that the system is improving.

Month	Cumulative number of defects	Month	Cumulative number of defects
1	17	31	2,217
2	39	32	2,430
3	53	33	2,586
4	87	34	3,884
5	106	35	4,099
6	140	36	4,385
7	165	37	5,104
8	286	38	8,074
9	359	39	10,120
10	412	40	12,618
11	461	41	16,715
12	555	42	21,606
13	654	43	24,592
14	747	44	27,789
15	836	45	29,739
16	926	46	30,843
17	989	47	32,011
18	1,049	48	32,599
19	1,103	49	33,010
20	1,152	50	33,707
21	1,182	51	34,103
22	1,213	52	34,426
23	1,225	53	34,736
24	1,266	54	34,903
25	1,306	55	35,110
26	1,331	56	35,261
27	1,363	57	35,440
28	1,443	58	35,614
29	1,495	59	35,763
30	1,737	60	35,876

TABLE I  
SOFTWARE FAILURE DATA.

In all the experiments, we used a probability of type I error  $\alpha = 0.01$ . The value of  $\gamma$  was set to  $1 - \alpha$ .

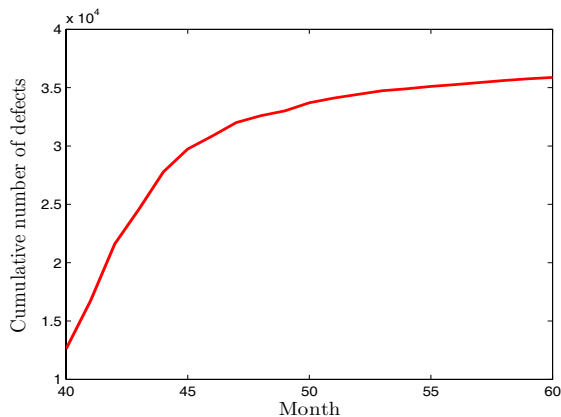


Fig. 4. Cumulative Number of Failures vs. Failure Time (DS II)

### 5.1 Qualitative evaluation of the proposed method

In this subsection, we present simulation results where the Bayesian prediction method [2], the Bayesian prediction using MCMC [18], OC curve approach, and the POC curve algorithm are applied to the software failure dataset (DS I) and also to the truncated software failure data (DS II).

For the Bayesian prediction method, the estimate of the parameter  $\hat{b}$  is equal to 0.3374, and for the Bayesian predic-

tion approach with MCMC the estimate of the value of  $\hat{b}$  is equal to 0.5402.

Fig. 5 and Fig. 6 show the prediction results of the proposed POC curve in comparison the Bayesian approaches for both datasets DS I and DS II respectively. These results clearly indicate that our method outperforms the Bayesian techniques used for comparison. Moreover, the proposed method is simple and easy to implement. One main advantage of the proposed algorithm is the nearly perfect fit between the predicted data and the observed data.

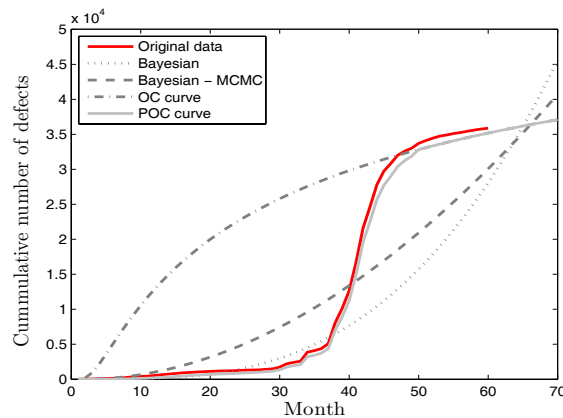


Fig. 5. Comparison of the prediction results for DS I.

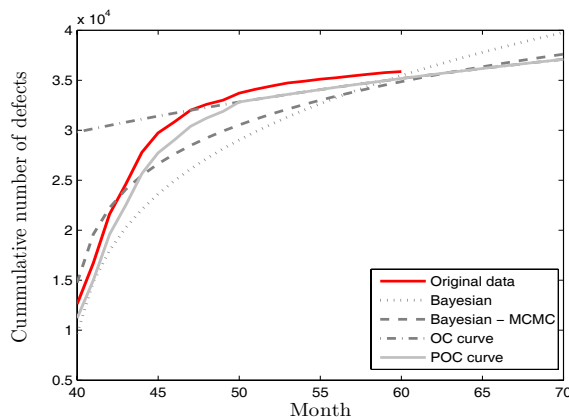


Fig. 6. Comparison of the prediction results for DS II.

### 5.2 Quantitative evaluation of the proposed method

Denote by  $N_o(t)$  and  $N_p(t)$  the observed and the predictive cumulative number of failures respectively.

To quantify the better performance of the proposed predictive method in comparison with the Bayesian approaches, we computed three goodness-of-fit measures: the skill score, the Nash-Sutcliffe model efficiency coefficient, and the relative error between the observed  $T_o \times 2$  data matrix

$$\mathcal{D}_o = \{(t, N_o(t)) : t = 1, \dots, T_o\},$$

and the predicted  $T_p \times 2$  data matrix

$$\mathcal{D}_p = \{(t, N_p(t)) : t = 1, \dots, T_p\}.$$

Note that the size of observed data matrix  $\mathcal{D}_o$  may not be equal to the size of the predicted data matrix  $\mathcal{D}_p$ , and hence an intersection step is necessary to pair up the observed data to the predicted data. This intersection function is setup to pair up the first column in the observed data matrix and the first column in the predicted data matrix. Data values are located in the second column of both matrices. More precisely, we create a subset of matched data  $\mathcal{D}_m = \{t, N_o(t), N_p(t) : t = 1, \dots, T_m\}$  that would be used to compute the following goodness-of-fit measures:

- 1) **Skill Score**: it is a error statistic that is used to quantify the accuracy of prediction models, and it defined as follows

$$SS = 1 - \frac{\sqrt{\frac{1}{T_m} \sum_{t=1}^{T_m} (N_o(t) - N_p(t))^2}}{\sqrt{\frac{1}{T_m-1} \sum_{t=1}^{T_m} (N_o(t) - \bar{N}_o)^2}},$$

The model prediction is better, when the value of the skill score  $SS$  is closer to one. When  $SS$  is less than zero, the model predictions are poor and the model errors are greater than observed data variability.

- 2) **Nash-Sutcliffe model efficiency coefficient**: is an indicator of the model's ability to predict about the 1:1 line between the observed and the predicted data, and it is defined as follows

$$E = 1 - \frac{\sum_{t=1}^{T_m} (N_o(t) - N_p(t))^2}{\sum_{t=1}^{T_m} (N_o(t) - \bar{N}_o)^2}.$$

The Nash-Sutcliffe model efficiency coefficient is a statistic similar to the skill score in that the closer to one the better the model prediction. A value of  $E = 1$  indicates that the model prediction is perfect, and if the value of  $E$  is equal to or less than zero, then the model prediction is considered poor.

- 3) **Relative error**: it measures how close a model is estimated with respect to the actual data. The relative error (RE) is defined as

$$RE = \frac{N_p(t) - N_o(t)}{N_o(t)}, \quad t = 1, \dots, T_m$$

The values of the three goodness-of-fit measures for all the experiments are depicted in Fig. 7 through Fig. Fig. 12, which clearly show that the proposed method gives the best results indicating the consistency with the subjective comparison.

## 6 Conclusions

In this paper, we introduced a new method for software defects prediction using operating characteristic curves. The core idea behind our proposed technique is to reliably predict the cumulative number of defects at any given stage during the software development process. The prediction accuracy of the proposed approach is validated on a real software failure data using several goodness-of-fit measures.

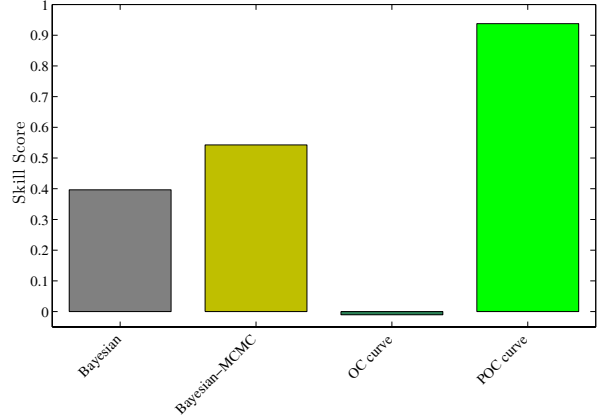


Fig. 7. Skill score results for DS I.

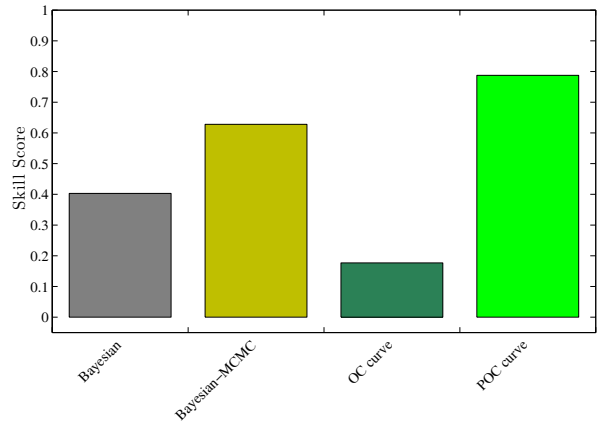


Fig. 8. Skill score results for DS II.

The experimental results clearly show a much improved performance of the proposed approach in comparison with the Bayesian prediction methods.

## Acknowledgments

This work was supported by SAP Labs Canada Inc.

## References

- [1] J.D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill Book Company, 1987.
- [2] J.W. Yu, G.L. Tian, and M.L. Tang, "Predictive analyses for nonhomogeneous Poisson processes with power law using Bayesian approach," *Computational Statistics & Data Analysis*, 2007.
- [3] C.G. Bai, "Bayesian network based software reliability prediction with an operational profile," *Journal of Systems and Software*, vol. 77, no. 2, pp. 103-112, 2004.
- [4] X. Zhang and H. Pham, "Software field failure rate prediction before software deployment," *Journal of Systems and Software*, vol. 79, pp. 291-300, 2006.

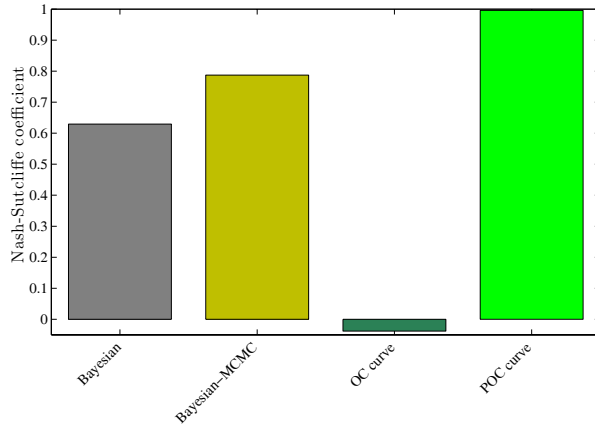


Fig. 9. Nash-Sutcliffe model efficiency coefficient results for DS I.

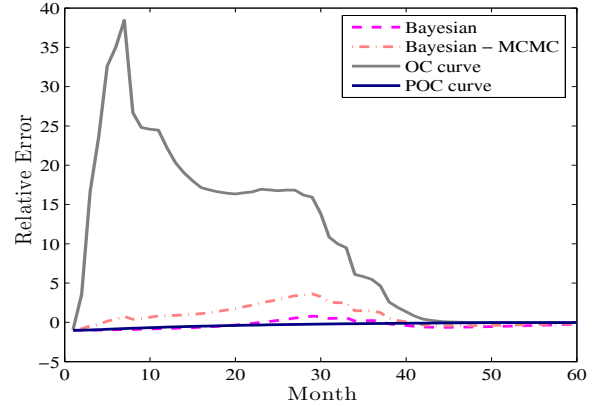


Fig. 11. Relative error results for DS I.

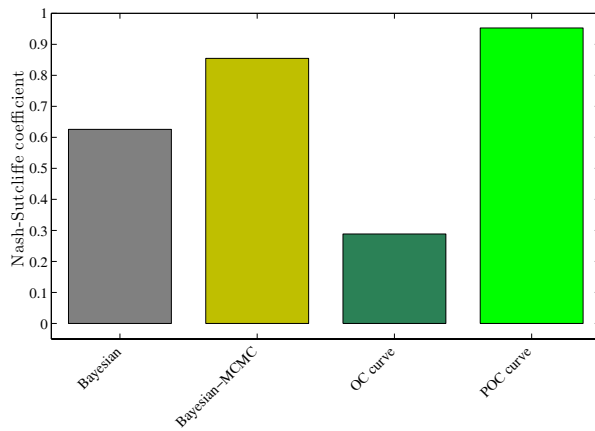


Fig. 10. Nash-Sutcliffe model efficiency coefficient results for DS II.

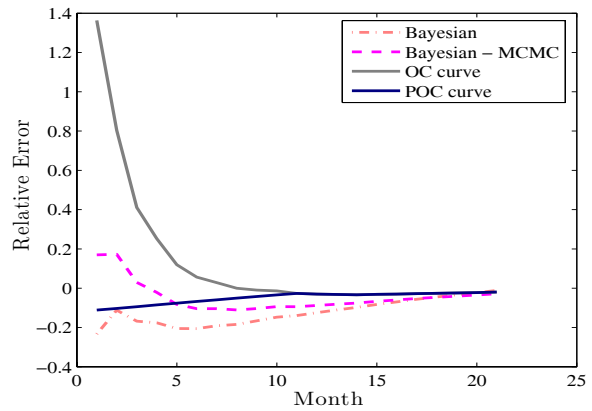


Fig. 12. Relative error results for DS II.

- [5] N.E. Fenton and M. Neil, "A critique of software defect prediction models," *IEEE Transactions on Software Engineering*, vol.5, no. 5, pp. 675-689, 1999.
- [6] J.D. Musa, "A theory of software reliability and its application," *IEEE Transactions on Software Engineering*, vol. 1, no. 1, pp. 312-327, 1975.
- [7] A.L. Goel and K. Okumoto, "Time-dependent error detection rate models for software reliability and other performance measures," *IEEE Transactions on Reliability*, vol. 28, no. 3, pp. 206-211, 1979.
- [8] M.R. Bastos Martini, K. Kanoun, and J. Moreira de Souza, "Software-reliability evaluation of the TROPICO-R switching system," *IEEE Transactions on Reliability*, vol. 39, no. 3, pp. 369-379, 1990.
- [9] K. Kanoun and J.C. Laprie, "Software reliability trend analysis from theoretical to practical considerations," *IEEE Transactions on Software Engineering*, vol. 41, no. 4, pp. 525-532, 1992.
- [10] A.L. Goel, "Software reliability models: assumptions, limitations and applicability," *IEEE Transactions on Software Engineering*, vol. 11, no. 12, pp. 1411-1423, 1985.
- [11] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Transactions on Reliability*, vol. 32, no. 5, pp. 475-485, 1983.
- [12] J.H. Lo and C.Y. Huang, "An integration of fault detection and correction processes in software reliability analysis," *Journal of Systems and Software*, vol. 79, no. 9, pp. 1312-1323, 2006.
- [13] O. Gauodin, "Optimal properties of the Laplace trend test for software-reliability models," *IEEE Transactions on Reliability*, vol. 20, no. 9, pp. 740-747, 1992.
- [14] H.E. Ascher and C.K.Hansen, "Spurious exponentiality observed when incorrectly fitting a distribution to non-stationary data," *IEEE Transactions on Reliability*, vol. 47, no. 4, pp. 451-45, 1998.
- [15] W.M. Bolstad, *Introduction to Bayesian Statistics*, John Wiley, 2004.
- [16] D.C. Montgomery, *Introduction to Statistical Quality Control*, John Wiley & Sons, 2005.
- [17] W.R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in Practice*, Chapman & Hall/CRC, 1995.
- [18] C. Robert, *Bayesian Choice*, 2nd Edition, Springer Verlag, NY, 2001.