The Journal of Systems & Software 191 (2022) 111367

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Burr-type NHPP-based software reliability models and their applications with two type of fault count data ☆

Siqiao Li *, Tadashi Dohi, Okamura Hiroyuki

Graduate School of Advanced Science and Engineering, Hiroshima University, 1-3-2 Kagamiyama, Higashi-Hiroshima City, Hiroshima, 739-8511, Japan

ARTICLE INFO

Article history:
Received 8 October 2021
Received in revised form 4 April 2022
Accepted 11 May 2022
Available online 21 May 2022

Keywords:
Software reliability models
Non-homogeneous Poisson processes
Burr-type distributions
Goodness-of-fit performance
Predictive performance

ABSTRACT

In this paper, we summarize the so-called Burr-type software reliability models (SRMs) based on the non-homogeneous Poisson process (NHPP) and comprehensively evaluate the model performances by comparing them with the existing NHPP-based SRMs. Two kinds of software fault count data are considered; fault-detection time-domain data and fault-detection time-interval data (group data). For 8 data sets in each fault count type, we estimate the model parameters by means of the maximum likelihood estimation and evaluate the performance metrics in terms of goodness-of-fit and prediction. It is shown that the Burr-type NHPP-based SRMs could show the better performances than the existing NHPP-based SRMs in many cases. The main contribution of the paper consists in suggesting that the Burr-type NHPP-based SRMs should be the possible candidates for selecting the best SRM in terms of goodness-of-fit and predictive performances.

1. Introduction

In the typical waterfall development model, the software development process consists of 5 steps: (i) requirement/specification analysis, (ii) preliminary and detailed design, (iii) coding, (iv) testing/verification, and (v) maintenance. In the testing phase especially, software faults are detected and removed as much as possible to meet high software reliability requirements. In other words, the success of software testing leads to guaranteeing the quality of software. Since software reliability is considered as one of the most fundamental and significant attributes of software quality, considerable attention has been paid to improving software testing. At the same time, since software testing is quite expensive, the quantification of software reliability is also another important issue in the verification phase. Since quantitative software reliability is defined as the probability that software failures caused by faults do not occur in a given time interval after the release, it is common to describe the probabilistic behavior of the fault-detection process in testing phases by any stochastic counting process. The software reliability defined in the above cannot be measured directly in the field, so that stochastic models, which are called *software reliability models* (SRMs), can be utilized to assess the quantitative software reliability. In fact, a great number of SRMs have been developed to control/monitor software testing processes as well as to evaluate the quantitative software reliability during the last four decades (Lyu, 1996; Musa et al., 1987).

It is well known that non-homogeneous Poisson process (NHPP)-based SRMs have been widely used to describe the behavior of the cumulative number of software faults. The representative NHPP-based SRMs are characterized by the mean value functions, which are proportional to the cumulative distribution functions (CDF) of software fault-detection time. Since the seminal contribution by Goel and Okumoto (1979), many authors proposed NHPP-based SRMs under different model assumptions. The representative NHPP-based SRMs assumed the exponential CDF (Goel and Okumoto, 1979), the gamma CDF (Yamada et al., 1983; Zhao and Xie, 1996), the truncated-logistic CDF (Ohba, 1984), the log–logistic CDF (Gokhale and Trivedi, 1998), the Pareto CDF (Abdel-Ghaly et al., 1986), the truncated-normal CDF (Okamura et al., 2013), the log-normal CDF (Okamura et al., 2013; Achcar et al., 1998), the extreme-value CDFs (Ohishi et al., 2009) including the Weibull CDF (Goel, 1985).

It is worth noting that the above CDFs are the representative lifetime distribution functions to model the time to failure in reliability engineering. On one hand, up to the present stage, we have known that no unique SRM, which could fit every software fault count data, was found yet, and that the best SRM strongly depended on the kind of software fault count data. Hence, one research direction was to provide a general modeling framework

☆ This paper is an extended version of the conference paper Li et al. (2021) presented by the same authors.

* Correspondence to: Dependable Systems Laboratory, 1-3-2 Kagamiyama, Higashi-Hiroshima City, Hiroshima, 739-8511, Japan.
E-mail addresses: rel-siqiao@hiroshima-u.ac.jp (S. Li), dohi@hiroshima-u.ac.jp (T. Dohi), okamu@hiroshima-u.ac.jp (O. Hiroyuki).

https://doi.org/10.1016/j.jss.2022.111367
0164-1212/

to describe the software fault-detection process. Langberg and Singpurwalla (1985), Miller (1986), Chen and Singpurwalla (1997) showed that the existing SRMs could be categorized into the Bayesian model, order statistics model, and self-exciting point process, respectively. Focusing on the NHPP-based SRMs, Gokhale and Trivedi (1999) proposed a coverage based-NHPP to interpret the CDF of software fault-detection time. Huang et al. (2003) found that the mean value function of NHPP-based SRMs can be represented by several kinds of algebraic mean operators. Xiao et al. (2012) gave a unified modeling framework of the exponentially shaped mean value function by introducing the equilibrium distribution. Okamura and Dohi (2016) developed another unified approach to approximate the CDF of software fault-detection time by the phase-type distributions.

However, it is emphasized that the unification approach does not always resolve the model selection problem because it never suggests which SRM is best in terms of goodness-of-fit and predictive performances. In other words, the model selection from the parametric forms such as Goel and Okumoto (1979), Yamada et al. (1983), Zhao and Xie (1996), Ohba (1984), Gokhale and Trivedi (1998), Abdel-Ghaly et al. (1986), Okamura et al. (2013), Achcar et al. (1998) and Ohishi et al. (2009) is still needed to determine the best SRM in the actual software reliability management, where the underlying fault-detection time belongs to a generalized exponential family or the extreme-value distribution family. The primary purpose of this paper is to summarize the so-called Burr-type SRMs based on the NHPP and evaluate the model performances comprehensively by comparing them with the existing NHPP-based SRMs. We assume the Burr-type III, VI, VII, VIII, IX, X, and XII distributions to describe the software fault-detection time distribution and compare the goodness-of-fit and predictive performances with the well-known 11 SRMs (Okamura and Dohi, 2013).

The rest of this paper is organized as follows. Section 2 gives the definition of NHPP-based SRM and the parameter estimation based on the maximum likelihood method, where two kinds of software fault count data are considered; fault-detection time-domain data and fault-detection time-interval data (group data). Section 3 introduces the definition of the Burr-type distributions and their associated NHPP-based SRMs, where almost all of them are newly proposed in this paper. Section 4 is devoted to numerical examples to compare our Burr-type NHPP-based SRMs with the existing ones. For 8 data sets in each fault count type, we estimate the model parameters by utilizing the maximum likelihood estimation and evaluate the performance metrics in terms of goodness-of-fit and prediction. We also assess the quantitative software reliability with all the NHPP-based SRMs and, compare the results. In Section 5, we summarize the related work on the Burr-type SRMs. Finally, the paper is concluded with some remarks in Section 6.

2. NHPP-based software reliability modeling

2.1. Non-homogeneous Poisson processes

Suppose that the testing phase of a software development project starts at time $t = 0$. Let $\{N(t), t \geq 0\}$ be a stochastic counting process to describe the cumulative number of software faults detected by time t (≥ 0). The stochastic process $N(t)$ is said a non-homogeneous Poisson process (NHPP) if the following conditions are satisfied:

- $N(0) = 0$,
- $\{N(t), t \geq 0\}$ has independent increment,
- $\Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t)$,
- $\Pr\{N(t + \Delta t) - N(t) = 1\} = \lambda(t; \boldsymbol{\theta})\Delta t + o(\Delta t)$,

where the function $\lambda(t; \boldsymbol{\theta})$ is an absolutely continuous (deterministic) function, called the *intensity function*, $\boldsymbol{\theta}$ is the model parameter (vector), and $o(\Delta t)$ indicates the higher-order term of the infinitesimal time Δt, which is given by

$$\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0. \tag{1}$$

If $N(t)$ follows an NHPP, the state transition probability, $P_n(t) = \Pr\{N(t) = n | N(0) = 0\}$, which is equivalent to the probability mass function (PMF), satisfies the Kolmogorov forward equations;

$$\frac{d}{dt} P_0(t) = -\lambda(t; \boldsymbol{\theta}) P_0(t), \tag{2}$$

$$\frac{d}{dt} P_n(t) = \lambda(t; \boldsymbol{\theta}) P_{n-1}(t) - \lambda(t; \boldsymbol{\theta}) P_n(t), \; n = 1, 2, \ldots. \tag{3}$$

Given the initial conditions; $P_0(0) = 1$ and $P_n(0) = 0$ $(n = 1, 2, \ldots)$, we immediately obtain

$$P_n(t) = \frac{\{\Lambda(t; \boldsymbol{\theta})\}^n}{n!} \exp(-\Lambda(t; \boldsymbol{\theta})) \; (n = 0, 1, 2, \ldots). \tag{4}$$

From the Poisson nature, we have

$$\mathrm{E}[N(t)] = \sum_{n=0}^{\infty} n P_n(t) = \Lambda(t; \boldsymbol{\theta}) = \int_0^t \lambda(x; \boldsymbol{\theta}) dx, \tag{5}$$

which is called the *mean value function* and denotes the expected cumulative number of software faults by time t.

2.2. The existing NHPP-based SRMs

It is assumed that each software fault is detected at independent and identically distributed (i. i. d.) random time with a non-degenerate cumulative distribution function (CDF), $F(t; \boldsymbol{\alpha})$, having the parameter $\boldsymbol{\alpha}$, and that the residual number of software faults at time $t = 0$ is a Poisson distributed random variable with parameter ω (> 0). Then the resulting software fault detection process obeys the NHPP with mean value function $\Lambda(t; \boldsymbol{\theta}) = \omega F(t; \boldsymbol{\alpha})$ with $\boldsymbol{\theta} = (\omega, \boldsymbol{\alpha})$. In this way, the commonly used assumption in software reliability engineering is that the initial number of residual software faults in a software system is expected to be finite, *i.e.*, $\lim_{t \to \infty} \Lambda(t; \boldsymbol{\theta}) = \omega \; (> 0)$.

In the classical software reliability modeling, the main research issue was to determine the intensity function $\lambda(t; \boldsymbol{\theta})$ or equivalently the mean value function $\Lambda(t; \boldsymbol{\theta})$ to fit the software fault count data. Okamura and Dohi (2013) implemented the existing NHPP-based SRMs with 11 software fault-detection time CDFs in the software reliability assessment tool on the spreadsheet (SRATS), which includes exponential (exp), gamma, Pareto, log-normal (lnorm), log–logistic (llogist), log-extreme-value minimum (lxvmin), log-extreme-value maximum (lxvmax), truncated logistic (tlogist), truncated normal (tnorm), truncated extreme-value minimum (txvmin), truncated extreme-value maximum (txvmax) distributions. In Table 1, we summarize these 11 NHPP-based SRMs.

2.3. Parameter estimation

Maximum likelihood (ML) estimation is a commonly used technique for the parameter estimation of NHPP-based SRMs. In ML estimation, the estimates are given by the parameters maximizing the log-likelihood function (LLF). On the other hand, the LLF value depends on the observed data as well as the underlying NHPP-based SRMs. In this paper, two types of data; time-domain data and group data, are considered.

Table 1
The existing NHPP-based SRMs.

Models	$\Lambda(t;\boldsymbol{\theta})$
Exponential dist. (exp) (Goel and Okumoto, 1979)	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = 1 - e^{-bt}$
Gamma dist. (gamma) (Yamada et al., 1983; Zhao and Xie, 1996)	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \int_0^t \frac{c^b s^{b-1} e^{-cs}}{\Gamma(b)} ds$
Pareto dist. (pareto) (Abdel-Ghaly et al., 1986)	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = 1 - (\frac{b}{t+b})^c$
Truncated normal dist. (tnorm) (Okamura et al., 2013)	$\Lambda(t;\boldsymbol{\theta}) = \omega \frac{F(t;\boldsymbol{\alpha}) - F(0;\boldsymbol{\alpha})}{1-F(0;\boldsymbol{\alpha})}$ $F(t;\boldsymbol{\alpha}) = \frac{1}{\sqrt{2\pi}b}\int_{-\infty}^t e^{-\frac{(s-c)^2}{2b^2}} ds$
Log-normal dist. (lnorm) (Achcar et al., 1998; Okamura et al., 2013)	$\Lambda(t;\boldsymbol{\theta}) = \omega F(\ln t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \frac{1}{\sqrt{2\pi}b}\int_{-\infty}^t e^{-\frac{(s-c)^2}{2b^2}} ds$
Truncated logistic dist. (tlogist) (Ohba, 1984)	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \frac{1-e^{-bt}}{1+ce^{-bt}}$
Log–logistic dist. (llogist) (Gokhale and Trivedi, 1998)	$\Lambda(t;\boldsymbol{\theta}) = \omega F(\ln t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \frac{(bt)^c}{1+(bt)^c}$
Truncated extreme-value max dist. (txvmax) (Ohishi et al., 2009)	$\Lambda(t;\boldsymbol{\theta}) = \omega \frac{F(t;\boldsymbol{\alpha}) - F(0;\boldsymbol{\alpha})}{1-F(0;\boldsymbol{\alpha})}$ $F(t;\boldsymbol{\alpha}) = e^{-e^{-\frac{t-c}{b}}}$
Log-extreme-value max dist. (lxvmax) (Ohishi et al., 2009)	$\Lambda(t;\boldsymbol{\theta}) = \omega F(\ln t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = e^{-e^{-\frac{t-c}{b}}}$
Truncated extreme-value min dist. (txvmin) (Ohishi et al., 2009)	$\Lambda(t;\boldsymbol{\theta}) = \omega \frac{F(0;\boldsymbol{\alpha}) - F(t;\boldsymbol{\alpha})}{F(0;\boldsymbol{\alpha})}$ $F(t;\boldsymbol{\alpha}) = e^{-e^{-\frac{t-c}{b}}}$
Log-extreme-value min dist. (lxvmin) (Goel, 1985)	$\Lambda(t;\boldsymbol{\theta}) = \omega\,(1 - F(-\ln t;\boldsymbol{\alpha}))$ $F(t;\boldsymbol{\alpha}) = e^{-e^{-\frac{t-c}{b}}}$

$(\omega > 0, a > 0, b > 0, c > 0)$

(i) Time-domain data

A set of fault detection times measured with CPU time is called the (fault-detection) time-domain data. Suppose that m_T software faults are detected, where the time sequence is given by $\boldsymbol{T} = \{t_1, t_2, \ldots, t_{m_T}\}$. Then, the likelihood function is represented as

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{T}) = \exp(-\Lambda(t_{m_T};\boldsymbol{\theta}))\prod_{i=1}^{m_T}\lambda(t_i;\boldsymbol{\theta}) \tag{6}$$

log-likelihood function is written by

$$\ln\mathcal{L}(\boldsymbol{\theta};\boldsymbol{T}) = \sum_{i=1}^{m_T}\ln\lambda(t_i;\boldsymbol{\theta}) - \Lambda(t_{m_T};\boldsymbol{\theta}). \tag{7}$$

By maximizing $\ln\mathcal{L}(\boldsymbol{\theta};\boldsymbol{T})$ with respect to $\boldsymbol{\theta}$, we seek the ML estimate $\hat{\boldsymbol{\theta}}$.

(ii) Group data

A group data consists of the number of faults detected in fixed time intervals measured with the calendar time, $(t_{i-1}, t_i]$ $(i = 1, 2, \ldots, m_G)$. Each record of the group data (t_i, n_i) is given by a pair of the observation time t_i and the cumulative number of software faults detected by time t_i. Then, the likelihood function and log-likelihood function with the group data $\boldsymbol{I} = \{(t_i, n_i), i = 1, 2, \ldots, m_G\}$ are given by

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{I}) = \prod_{i=1}^{m_G}\left[\frac{[\Lambda(t_i;\boldsymbol{\theta}) - \Lambda(t_{i-1};\boldsymbol{\theta})]^{n_i-n_{i-1}}}{(n_i - n_{i-1})!}\right] \times e^{-[\Lambda(t_i;\boldsymbol{\theta}) - \Lambda(t_{i-1};\boldsymbol{\theta})]}, \tag{8}$$

$$\ln\mathcal{L}(\boldsymbol{\theta};\boldsymbol{I}) = \sum_{i=1}^{m_G}(n_i - n_{i-1})\ln\{\Lambda(t_i;\boldsymbol{\theta}) - \Lambda(t_{i-1};\boldsymbol{\theta})\} - \sum_{i=1}^{m_G}\ln\{(n_i - n_{i-1})!\} - \lambda(t_{m_G};\boldsymbol{\theta}), \tag{9}$$

respectively, where $(t_0, n_0) = (0, 0)$. Hence the ML estimate $\hat{\boldsymbol{\theta}}$ is given by the solution of $\mathrm{argmax}_{\boldsymbol{\theta}}\ln\mathcal{L}(\boldsymbol{\theta};\boldsymbol{I})$.

3. Burr-type NHPP-based SRMs

For a continuous random variable X with the support $(-\infty, +\infty)$, let $F(x;\boldsymbol{\alpha})$ and $f(x;\boldsymbol{\alpha})$ be the CDF and the probability density function (PDF), respectively, where $F(x;\boldsymbol{\alpha})$ is an absolutely continuous non-decreasing function from $F(-\infty;\boldsymbol{\alpha}) = 0$ to $F(\infty;\boldsymbol{\alpha}) = 1$. For arbitrary a and b $(a < b)$, $\Pr\{a \le X \le b\} = F(b;\boldsymbol{\alpha}) - F(a;\boldsymbol{\alpha}) = \int_a^b f(x;\boldsymbol{\alpha})dx$ with $F(x;\boldsymbol{\alpha}) = \int_{-\infty}^x f(x;\boldsymbol{\alpha})dx$ and $f(x;\boldsymbol{\alpha}) = dF(x;\boldsymbol{\alpha})/dx$. Burr (1942) introduced a new family of CDFs which satisfy the following differential equation;

$$\frac{dF(x;\boldsymbol{\alpha})}{dx} = F(x;\boldsymbol{\alpha})(1 - F(x;\boldsymbol{\alpha}))g(x, F(x;\boldsymbol{\alpha})), \tag{10}$$

where $g(x, F(x;\boldsymbol{\alpha}))$ is an arbitrary positive function with $0 \le F(x;\boldsymbol{\alpha}) \le 1$. If $g(x, F(x;\boldsymbol{\alpha})) = \left(b_1 + b_2 x + b_3 x^2\right)^{-1}$ and if $F(x;\boldsymbol{\alpha})$ and $1 - F(x;\boldsymbol{\alpha})$ are replaced by $f(x)$ and $(b_0 - x)$, respectively, with arbitrary constants b_0, b_1, b_2, and b_3, then Eq. (10) is reduced to the differential equation for the well-known Pearson system;

$$\frac{df(x;\boldsymbol{\alpha})}{dx} = \frac{f(x;\boldsymbol{\alpha})(b_0 - x)}{\left(b_1 + b_2 x + b_3 x^2\right)}, \tag{11}$$

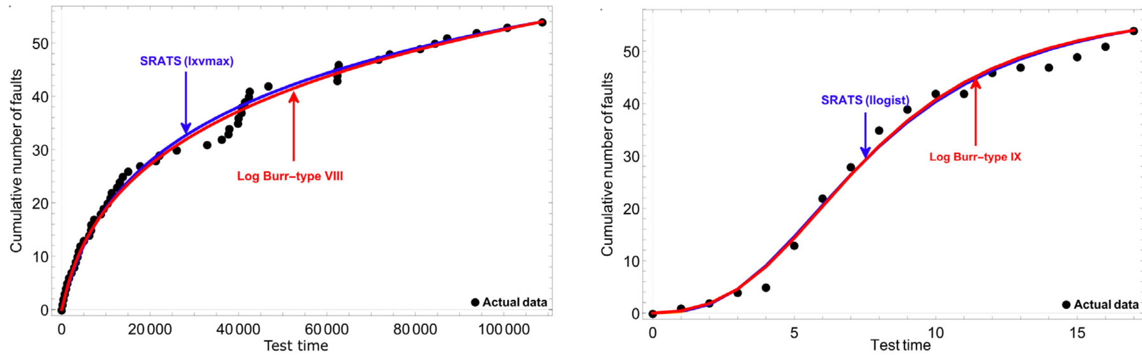
which leads to many popular CDFs, such as Pearson-type I (beta distribution), Pearson-type III (gamma distribution), Pearson-type VIII (power distribution), Pearson-type X (exponential distribution) and Pearson-type XI (a particular class of Pareto distribution).

Burr (1942) considered a special case of $g(x, F(x;\boldsymbol{\alpha})) = g(x;\boldsymbol{\alpha})$. By solving Eq. (10), we obtain

$$F(x;\boldsymbol{\alpha}) = \frac{1}{\left[e^{-\int g(x;\boldsymbol{\alpha})dx} + 1\right]}. \tag{12}$$

It should be noted that the selection of the function $g(x;\boldsymbol{\alpha})$ makes the CDF $F(x;\boldsymbol{\alpha})$ increase monotonously from 0 to 1 within a specified time x. The above statement is often called the *Burr hypothesis*. Finally, Burr (1942) derived 12 Burr-type distributions I∼XII by considering 12 kinds of $g(x;\boldsymbol{\alpha})$ functions. Table 2 lists the Burr-type distributions proposed in Burr (1942).

The Burr-type III, X, and XII distributions were applied to describe the software fault-detection time distribution in the past literature (see Section 5), where these CDFs have positive support $(0, \infty)$. In other words, from , it is immediate to see that the Burr-type I, IV, V, and XI distributions are not appropriate in modeling the software fault-detection time. In addition to the Burr-type III distribution (Ahmad et al., 2009, 2011; Sobhana and Prasad, 2015; Chowdary et al., 2015; Sridevi and Rani, 2015), the Burr-type X distribution (Sridevi and Akbar, 2016), the Burr-type XII distribution (Abdel-Ghaly et al., 1997; An, 2012; Kim and Park, 2010; Kim, 2013; Prasad et al., 2014b,c,a; Ravikumar and Kantam, 2017; Islam, 2020) with the positive support $X \in (0, \infty)$, it is possible to transform the CDF with support $(-\infty, +\infty)$ to the log Burr-type distributions and the truncated Burr-type distributions with the support $X \in (0, \infty)$ by taking $\exp(X)$ and truncating X at the origin, respectively. So, we consider the log Burr-type II, VI, VII, VIII, IX distributions and the truncated Burr-type II, VI, VII, VIII, IX distributions to represent the mean value function of the



(a) TDS1. (b) GDS1.

Fig. 1. Behavior of cumulative number of software faults with the Burr-type and existing SRMs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Burr-type distributions.

Type	CDF	Domain of $\boldsymbol{x}$
I	$F(x;\boldsymbol{\alpha}) = x$	$(0, 1)$
II	$F(x;\boldsymbol{\alpha}) = (e^{-x}+1)^{-b}$	$(-\infty, +\infty)$
III	$F(x;\boldsymbol{\alpha}) = \left(1+(x)^{-a}\right)^{-b}$	$(0, +\infty)$
IV	$F(x;\boldsymbol{\alpha}) = \left(((c-x)/x)^{1/c}+1\right)^{-b}$	$(0, c)$
V	$F(x;\boldsymbol{\alpha}) = \left(ae^{-\tan x}+1\right)^{-b}$	$(-\pi/2, \pi/2)$
VI	$F(x;\boldsymbol{\alpha}) = \left(ae^{-c\sinh(x)}+1\right)^{-b}$	$(-\infty, +\infty)$
VII	$F(x;\boldsymbol{\alpha}) = 2^{-b}\left(1+\tanh(x)\right)^{b}$	$(-\infty, +\infty)$
VIII	$F(x;\boldsymbol{\alpha}) = (\arctan(e^{x})2/\pi)^{b}$	$(-\infty, +\infty)$
IX	$F(x;\boldsymbol{\alpha}) = 1-2\left(a\left((1+e^{x})^{b}-1\right)+2\right)^{-1}$	$(-\infty, +\infty)$
X	$F(x;\boldsymbol{\alpha}) = \left(1-e^{-(x)^{2}}\right)^{b}$	$(0, +\infty)$
XI	$F(x;\boldsymbol{\alpha}) = (x-(1/2\pi)\sin 2\pi x)^{b}$	$(0, 1)$
XII	$F(x;\boldsymbol{\alpha}) = 1-(1+x^{a})^{-b}$	$(0, +\infty)$

$(\omega > 0, a > 0, b > 0, c > 0)$

NHPP-based SRM by

$$\Lambda(t;\boldsymbol{\theta}) = \omega F(\ln t;\boldsymbol{\alpha}), \tag{13}$$

$$\Lambda(t;\boldsymbol{\theta}) = \omega\frac{F(t;\boldsymbol{\alpha})-F(0;\boldsymbol{\alpha})}{1-F(0;\boldsymbol{\alpha})}, \tag{14}$$

respectively. The underlying idea on the log Burr-type distribution comes from the log-normal NHPP-based SRM (Achcar et al., 1998; Okamura et al., 2013) and the log–logistic NHPP-based SRM (Gokhale and Trivedi, 1998). In fact, it is known that the logarithmic Burr-type II distribution is reduced to the log–logistic distribution (Tadikamalla, 1980). The truncation at the origin for the Burr II, VI, VII, VIII, IX distributions with the support $(-\infty, +\infty)$ is inspired by the truncated normal NHPP-based SRM (Okamura et al., 2013) and the truncated logistic NHPP-based SRM (Ohba, 1984).[1] Table 3 presents the Burr-type NHPP-based SRMs considered in this paper, where we applied generalized Burr-type III, VI, VII, VIII, IX, X, and XII distributions by introducing an additional scale parameter d. That is to say, if $d = 1$, then the Burr-type distributions in Table 3 become the original form in .

[1] The truncated Burr-type II, VI, VII, VIII, and IX NHPP-based SRMs are introduced in this paper, although the logarithmic Burr-type II, VI, VII, VIII, IX NHPP-based SRMs were proposed in the Li et al. (2021). In fact, we carry out again the comprehensive numerical experiments reported in Li et al. (2021) by adding the truncated Burr-type NHPP-based SRMs and show that these give the better goodness-of-fit and predictive performances in many cases.

4. Performance comparisons

4.1. Data sets

In numerical experiments, we analyze 8 software fault-detection time-domain data (TDS1∼TDS8) and 8 group data (GDS1∼GDS8) in Table 4. These were observed in actual software development processes and were analyzed in the past literature.

4.2. Goodness-of-fit performances

We investigate the goodness-of-fit of our 11 Burr-type NHPP-based SRMs and the existing 11 NHPP-based SRMs in SRATS (Okamura and Dohi, 2013). Based on the software fault counts experienced in the past, we seek the ML estimate $\hat{\boldsymbol{\theta}}$ and maximize the log-likelihood function $\ln\mathcal{L}(\boldsymbol{\theta};\boldsymbol{T})$ or $\ln\mathcal{L}(\boldsymbol{\theta};\boldsymbol{I})$. Then the Akaike information criterion (AIC) and the mean squares error (MSE) are defined by

$$\begin{aligned}\mathrm{AIC} = &-2\ln\mathcal{L}(\hat{\boldsymbol{\theta}};\boldsymbol{T}\text{ or }\boldsymbol{I})\\ &+2\times(\text{the number of parameters})\end{aligned} \tag{15}$$

and

$$\mathrm{MSE}(\hat{\boldsymbol{\theta}};\boldsymbol{T}) = \frac{\sum_{i=1}^{m_T}(i-\Lambda(t_i;\hat{\boldsymbol{\theta}}))^2}{m_T}, \tag{16}$$

$$\mathrm{MSE}(\hat{\boldsymbol{\theta}};\boldsymbol{I}) = \frac{\sum_{i=1}^{m_G}(n_i-\Lambda(t_i;\hat{\boldsymbol{\theta}}))^2}{m_G}, \tag{17}$$

where m_T and m_G are the total number of data and n_i is the cumulative number of software faults detected by time t_i. The smaller AIC/MSE is the better SRM in terms of the goodness-of-fit to the underlying fault count data.

Fig. 1 illustrates the mean value functions and the cumulative number of software faults detected in TDS1 and GDS1. The best SRMs with minimum AIC were selected from the 11 Burr-type NHPP-based SRMs (red curve) and the existing NHPP-based SRMs in SRATS (blue curve). At the first look, both modeling frameworks showed almost similar behavior. We present the best AIC results for the time-domain data and group data in Table 5 and Table 6 for more accurate comparisons, respectively, where the bold font marks the best SRM with minimum AIC in each data set. From Table 5, it can be seen that in 5 time-domain data sets (TDS1, TDS2, TDS4, TDS5, TDS8), our Burr-type NHPP-based SRMs could provide the better goodness-of-fit performances than the existing NHPP-based SRMs in SRATS. The MSE with the ML estimate was also compared as a distance metric between the mean value function and the underlying fault count data, while the AIC denotes an approximate distance between our assumed

Table 3
Burr-type NHPP-based SRMs.

Models	Mean value function
Burr-type III	$\Lambda(t;\boldsymbol{\theta}) = \omega\left(1+(t/d)^{-a}\right)^{-b}$
Log Burr-type VI	$\Lambda(\ln t;\boldsymbol{\theta}) = \omega\left(ae^{-c\sinh(\ln t/d)}+1\right)^{-b}$
Truncated (Tru)Burr-type VI	$\Lambda(t;\boldsymbol{\theta}) = \omega\left(\frac{\left(ae^{-c\sinh(t/d)}+1\right)^{-b}-\left(ae^{-c\sinh(0)}+1\right)^{-b}}{1-\left(ae^{-c\sinh(0)}+1\right)^{-b}}\right)$
Log Burr-type VII	$\Lambda(\ln t;\boldsymbol{\theta}) = \omega 2^{-b}\left(1+\tanh\left(\ln t/d\right)\right)^{b}$
Truncated (Tru)Burr-type VII	$\Lambda(t;\boldsymbol{\theta}) = \omega\left(\frac{2^{-b}(1+\tanh(t/d))^{b}-2^{-b}(1+\tanh(0))^{b}}{1-2^{-b}(1+\tanh(0))^{b}}\right)$
Log Burr-type VIII	$\Lambda(\ln t;\boldsymbol{\theta}) = \omega\left(\arctan(e^{\ln t/d})2/\pi\right)^{b}$
Truncated (Tru)Burr-type VIII	$\Lambda(t;\boldsymbol{\theta}) = \omega\left(\frac{\left(\arctan(e^{t/d})2/\pi\right)^{b}-(2/\pi\arctan(1))^{b}}{1-(2/\pi\arctan(1))^{b}}\right)$
Log Burr-type IX	$\Lambda(\ln t;\boldsymbol{\theta}) = \omega\left(1-2\left(a\left(\left(1+e^{\ln t/d}\right)^{b}-1\right)+2\right)^{-1}\right)$
Truncated (Tru)Burr-type IX	$\Lambda(t;\boldsymbol{\theta}) = \omega\left(\frac{\left(2^{b}+1\right)^{-1}-\left(\left(1+e^{t/d}\right)^{b}+1\right)^{-1}}{\left(2^{b}+1\right)^{-1}}\right)$
Burr-type X	$\Lambda(t;\boldsymbol{\theta}) = \omega\left(1-e^{-(t/d)^{2}}\right)^{b}$
Burr-type XII	$\Lambda(t;\boldsymbol{\theta}) = \omega\left(1-\left(\frac{1}{1+(t/d)^{a}}\right)^{b}\right)$

$(\omega > 0, a > 0, b > 0, c > 0, d > 0)$

Table 4
Data sets.

(a) Time data sets.

Data	No. faults	Source
TDS1	54	SYS2 (Musa, 1979)
TDS2	38	SYS3 (Musa, 1979)
TDS3	136	SYS1 (Musa, 1979)
TDS4	53	SYS4 (Musa, 1979)
TDS5	73	Project J5 (Lyu, 1996)
TDS6	38	S10 (Musa, 1979)
TDS7	41	S27 (Musa, 1979)
TDS8	101	S17 (Musa, 1979)

(b) Group data sets.

Data	No. faults	Testing periods	Source
GDS1	54	17 days	SYS2 (Musa, 1979)
GDS2	38	14 days	SYS3 (Musa, 1979)
GDS3	120	19 weeks	Release2 (Wood, 1996)
GDS4	61	12 weeks	Release3 (Wood, 1996)
GDS5	9	14 operational testing times	NASA -supported project (Vouk, 1992)
GDS6	66	20 days	DS1 (Okamura et al., 2011)
GDS7	58	33 days	DS2 (Okamura et al., 2011)
GDS8	52	30 days	DS3 (Okamura et al., 2011)

SRM and the real stochastic process behind the data. It is found that the Burr-type NHPP-based SRMs gave the smaller MSE than the existing NHPP-based SRMs in SRATS in TDS1, TDS2, TDS5, and TDS7 as well. In the group data sets, the Burr-type NHPP-based SRMs could not provide the smaller AIC in only GDS5 and GDS7, and the smaller MSE in only GDS7. Note that the significant difference in terms of AIC can be considered as greater than 2 from the definition of AIC. In the group data analysis, the remarkable differences between the Burr-type NHPP-based SRM and the existing NHPP-based SRM were not observed in GDS1, GDS3, GDS4, GDS5, GDS7, and GDS8. On the other hand, we notice that the difference in AIC between the best Burr-type NHPP-based SRM (Log Burr-type IX) and the best SRATS SRM (lxvmax) was greater than 2 in GDS2 and GDS6. These results reinforce the conclusion by Imanaka and Dohi Imanaka and Dohi (2015) so that the Burr-type NHPP-based SRM does not uniquely outperform the existing NHPP-based SRMs but provides better goodness-of-fit performances in many cases.

In Tables 5 and 6, we estimate the mean number of inherent software faults before software testing, $\hat{\omega}$, and calculate the absolute difference between the total number of software faults (m or n_m) and $\hat{\omega}$ as **diff**. The smaller **diff** implies a more plausible SRM under the assumption that no software fault was found after the release in all the data sets. The results tell us that even if some SRMs can guarantee the minimum AIC and MSE, the corresponding **diff** is not always minimized. In TDS3, TDS5, TDS7, and TDS8, the Burr-type NHPP-based SRMs could estimate the number of inherent faults more accurately than the existing NHPP-based SRMs in SRATS. In the group data sets, the Burr-type NHPP-based SRMs could also give more accurate estimates of the number of inherent faults than the SRATS SRMs in six cases (GDS1∼GDS6). Compared with the existing NHPP-based SRMs, we can conclude that our Burr-type NHPP-based SRMs are quite attractive in software reliability modeling and should be competitors with the high potential ability for the existing NHPP-based SRMs.

Table 5
Goodness-of-fit performances based on AIC (time-domain data).

Data Set	Burr Type				SRATS			
	Best Burr	AIC	$\hat{\omega}$(**diff**)	MSE	Best SRATS	AIC	$\hat{\omega}$(**diff**)	MSE
TDS1	Log Burr-type VIII	**896.663**	241.894 (187.894)	**1.941**	lxvmax	896.666	232.175 (**178.175**)	1.950
TDS2	Log Burr-type VIII	**598.122**	84.883 (46.883)	**1.674**	lxvmax	598.131	82.895 (**44.895**)	1.705
TDS3	Burr-type X	1939.258	143.726 (**4.726**)	13.321	lxvmin	**1938.160**	172.526 (34.526)	**6.570**
TDS4	Tru Burr-type IX	**759.454**	56.139 (3.139)	4.185	pareto	759.756	53.201 (**0.201**)	**3.784**
TDS5	Burr-type X	**757.119**	84.808(**11.808**)	**16.910**	exp	757.869	95.960 (22.960)	18.985
TDS6	Log Burr-type VIII	722.397	66.989(28.989)	2.095	lxvmax	**721.928**	51.561 (**13.561**)	**1.442**
TDS7	Log Burr-type VII	1008.220	95.007 (**54.007**)	**5.967**	lxvmax	1008.220	95.796(54.796)	5.970
TDS8	Tru Burr-type VI	**2502.160**	141.105(**40.105**)	50.351	pareto	2504.170	211.202 (110.202)	**47.404**

Table 6
Goodness-of-fit performances based on AIC (group data).

Data Set	Burr Type				SRATS			
	Best Burr	AIC	$\hat{\omega}$(**diff**)	MSE	Best SRATS	AIC	$\hat{\omega}$(**diff**)	MSE
GDS1	Log Burr-type IX	**72.500**	59.288 (**5.288**)	**3.819**	llogist	73.053	60.651 (6.651)	4.115
GDS2	Log Burr-type IX	**59.459**	67.468 (**29.459**)	**3.212**	lxvmax	61.694	74.334 (36.334)	3.239
GDS3	Log Burr-type VI	**85.873**	122.896 (**2.896**)	**3.086**	tnorm	87.267	123.252 (3.252)	6.151
GDS4	Tru Burr-type IX	**50.256**	61.797 (**0.797**)	**1.816**	tlogist	51.052	62.269 (1.269)	1.968
GDS5	Log Burr-type IX	30.231	30.359 (**21.359**)	**0.071**	exp	**29.911**	39.052 (30.052)	0.118
GDS6	Log Burr-type IX	**103.459**	71.982(**5.983**)	**20.890**	lxvmax	108.831	134.936 (68.936)	22.514
GDS7	Burr-type III	124.767	58.178 (0.178)	2.180	txvmin	**123.265**	58.037 (**0.037**)	**2.122**
GDS8	Log Burr-type IX	**117.234**	52.516 (0.516)	**9.042**	llogist	117.470	52.459 (**0.459**)	9.408

4.3. Predictive performances

It is worth mentioning that the better goodness-of-fit to the past observation does not always lead to the better performance for the future prediction. Since assessing the quantitative software reliability predicts the fault-free probability during a future testing/operational period, it is important to investigate the predictive performance of the Burr-type NHPP-based SRMs. The predictive performance is measured by the predictive mean squares error (PMSE) to evaluate the average squared distance between the predicted cumulative number of software faults and its (unknown) realization per prediction length. Suppose that m_T or n_{m_G} software fault counts data are available and that the prediction length is given by l $(= 1, 2, \ldots)$. Then, we define the PMSE for the fault-detection time-domain data and the group data by

$$\mathrm{PMSE}(\hat{\boldsymbol{\theta}}; \boldsymbol{T}) = \frac{\sum_{i=m_T+1}^{m_T+l} \{i - \Lambda(t_i; \hat{\boldsymbol{\theta}})\}^2}{l}, \tag{18}$$

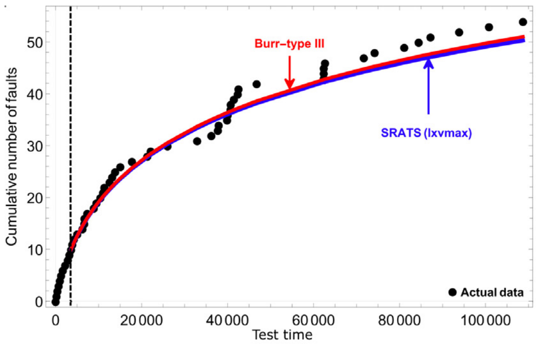
$$\mathrm{PMSE}(\hat{\boldsymbol{\theta}}; \boldsymbol{I}) = \frac{\sum_{i=m_G+1}^{m_G+l} \{n_i - \Lambda(t_i; \hat{\boldsymbol{\theta}})\}^2}{l}, \tag{19}$$

where $\hat{\boldsymbol{\theta}}$ is the ML estimate at time t_{m_T} or t_{m_G}.

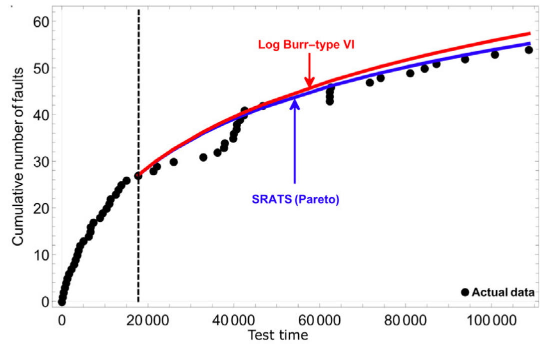
Our experiments set three observation points; 20%, 50%, and 80% of the whole data set, predict the cumulative number of software faults for the remaining period, say, 80%, 50%, and 20% lengths, and calculate the PMSEs in all the cases with all SRMs. The prediction length becomes shorter as the observation point is larger. In Figs. 2 and 3, we show the examples of predictive behavior of the cumulative number of software faults with the Burr-type and existing SRMs in TDS1 and GDS1, respectively, where the dotted line denotes the prediction point. In these figures, we plot the best predictive SRMs with the minimum PMSE. In Fig. 2, since the underlying fault-detection time behaves like an exponential curve, both SRMs; the Burr-type NHPP-based SRM and the SRATS SRM could show a similar prediction trend. On the other hand, the group data in Fig. 3 represents the S-shaped curve, and both SRMs resulted in the miss-prediction in the early testing phases, like 20% and 50% observation points. The trend change in the future causes these poor predictive performances. More specifically, In Fig. 3(a), both SRMs could not predict the S-shaped increasing trend. In Fig. 3(b), they failed to predict the 3 steps increasing trend. From these results, we can understand that the prediction of the future unknown trend change is essentially difficult, even though the prediction length is relatively short.

Tables 7 and 8 present the comparison results on the PMSE in time-domain data sets and group data sets, respectively, where we select the best SRM with the smallest PMSE from the Burr-type NHPP-based SRMs and the SRATS SRMs. In the time-domain data, it is seen that the Burr-type NHPP-based SRMs could guarantee the smaller PMSE than SRATS SRMs in half of the cases of early (20%) and middle (50%) testing phases. When the testing phase is later (80%), the Burr-type NHPP-based SRMs provided the smaller PMSE in 7 out of 8 cases. In the group data analysis, regardless of the prediction length, the Burr-type NHPP-based SRMs gave the better predictive performances than the SRATS SRMs in more than half of the data sets.

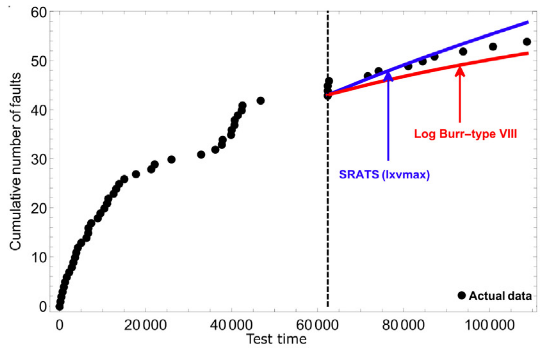
Except in the 80% observation of Table 7, it should be noted that the best SRM with the minimum PMSE depends on the data sets in both modeling frameworks; Burr-type and SRATS. Of course, the best SRM with the minimum PMSE cannot be known in advance at each observation point. In this sense, we have to say that the comparison in Tables 7 and 8 is not feasible at each prediction point. In Tables 9 and 10, we compare the predictive performances of SRMs with the minimum AIC at each observation point in the time-domain data sets and group data sets, respectively. In the time-domain data, we can observe that when the testing phase is early (20%), the existing NHPP-based SRMs could show the smaller AIC and smaller PMSE than the Burr-type NHPP-based SRMs in 6 out of 8 cases (TDS1, TDS2, TDS5, TDS6, TDS7, and TDS8), and the Log Burr-type IX SRM outperformed the SRATS SRMs in only TDS3. When the testing phase is middle (50%), Burr-type NHPP-based SRMs could provide the smaller AIC and smaller PMSE than existing NHPP-based SRMs in TDS7, and



(a) 20% observation point.

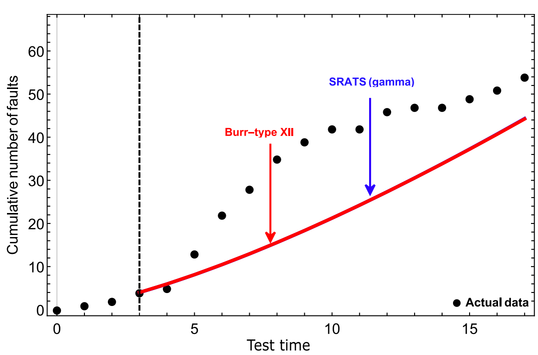


(b) 50% observation point.

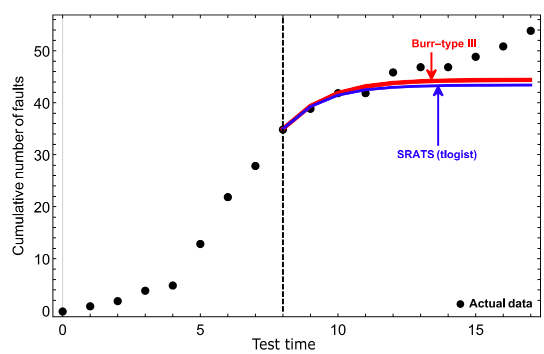


(c) 80% observation point.

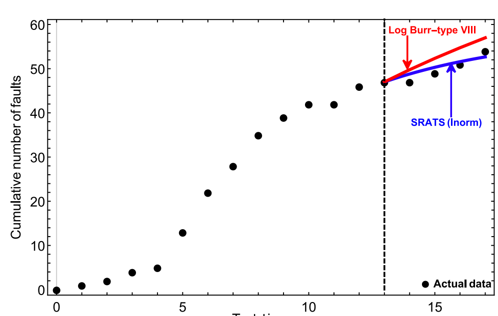
Fig. 2. Predictive behavior of cumulative number of software faults with the Burr-type and existing SRMs in TDS1.



(a) 20% observation point.



(b) 50% observation point.



(c) 80% observation point.

Fig. 3. Predictive behavior of cumulative number of software faults with the Burr-type and existing SRMs in GDS1.

the exponential NHPP-based SRM outperformed the Burr-type NHPP-based SRMs in TDS1 and TDS2. When the testing phase is later (80%), the Burr-type NHPP-based SRMs could show the best goodness-of-fit performance in the observation phase and ensure the minimum PMSE in the future prediction phase in TDS1, TDS3, TDS7, and TDS8.

In the group data of Table 10, it can be observed that the Burr-type NHPP-based SRMs provided both the smallest AIC and

Table 7
Predictive performances based on PMSE (time-domain data).

Data Set	Burr Type		SRATS	
	Best Burr	PMSE	Best SRATS	PMSE
(i) Prediction from the 20% observation point				
TDS1	Burr-type III	**3.951**	lxvmax	5.073
TDS2	Burr-type XII	270.380	tnorm	**42.104**
TDS3	Log Burr-type VI	**28.422**	lxvmax	32.131
TDS4	Burr-type III	59.027	lnorm	**56.477**
TDS5	Tru Burr-type VI	**2022.990**	exp	9177.670
TDS6	Burr-type XII	**51.861**	exp	84.035
TDS7	Log Burr-type VII	88.338	lxvmax	**32.217**
TDS8	Burr-type III	2401.850	lxvmax	**1852.520**
(ii) Prediction from the 50% observation point				
TDS1	Log Burr-type VI	9.009	pareto	**6.118**
TDS2	Tru Burr-type IX	**3.749**	tlogist	14.890
TDS3	Tru Burr-type IX	259.988	pareto	**11.712**
TDS4	Tru Burr-type VIII	121.671	tlogist	**103.504**
TDS5	Log Burr-type VIII	252.978	llogist	**193.903**
TDS6	Log Burr-type VIII	**4.358**	lxvmax	10.493
TDS7	Tru Burr-type IX	**37.702**	exp	4480.620
TDS8	Tru Burr-type VI	**194.841**	lxvmax	32375.500
(iii) Prediction from the 80% observation point				
TDS1	Log Burr-type VIII	**5.720**	lxvmax	5.772
TDS2	Log Burr-type VIII	**0.583**	lxvmax	0.588
TDS3	Tru Burr-type VI	**7.997**	lxvmax	9.419
TDS4	Log Burr-type IX	**2.867**	txvmin	4.253
TDS5	Tru Burr-type VI	32.079	lxvmax	**21.715**
TDS6	Burr-type III	**2.032**	lxvmax	2.041
TDS7	Burr-type XII	**9.161**	lxvmax	10.498
TDS8	Burr-type XII	**46.212**	lxvmax	57.901

Table 8
Predictive performances based on PMSE (group data).

Data Set	Burr Type		SRATS	
	Best Burr	PMSE	Best SRATS	PMSE
(i) Prediction from the 20% observation point				
GDS1	Burr-type XII	**219.067**	gamma	220.732
GDS2	Log Burr-type VI	**9.149**	lxvmax	29.244
GDS3	Log Burr-type IX	**429.795**	gamma	820.049
GDS4	Burr-type XII	791.335	exp	**142.854**
GDS5	Log Burr-type VII	3.535	pareto	**2.628**
GDS6	Log Burr-type IX	**41.897**	tlogist	98.903
GDS7	Tru Burr-type IX	552.996	exp	**387.694**
GDS8	Tru Burr-type VI	448.935	txvmin	**423.360**
(ii) Prediction from the 50% observation point				
GDS1	Burr-type III	**26.557**	tlogist	157.837
GDS2	Burr-type XII	45.793	txvmin	**30.786**
GDS3	Log Burr-type VIII	**346.721**	lxvmax	564.782
GDS4	Log Burr-type VIII	340.914	exp	**101.303**
GDS5	Log Burr-type IX	**0.300**	exp	0.344
GDS6	Log Burr-type IX	**327.310**	pareto	365.493
GDS7	Burr-type III	**22.561**	lxvmax	22.894
GDS8	Tru Burr-type IX	**20.613**	txvmin	29.110
(iii) Prediction from the 80% observation point				
GDS1	Log Burr-type VIII	4.676	lnorm	**1.762**
GDS2	Log Burr-type VI	**0.455**	exp	0.464
GDS3	Tru Burr-type IX	**0.230**	tnorm	0.331
GDS4	Tru Burr-type IX	**0.695**	tnorm	1.850
GDS5	Log Burr-type VI	**0.152**	tnorm	0.224
GDS6	Burr-type III	**1.710**	lnorm	3.432
GDS7	Burr-type X	21.403	txvmin	**6.118**
GDS8	Log Burr-type VIII	0.862	lxvmax	0.864

smallest PMSE at the same time in some cases; *i.e.*, one case out of 8 data sets in (i), 6 cases out of 8 data sets in (ii) and 5 cases out of 8 data sets in (iii). These results confirm that the Burr-type NHPP-based SRMs have the higher prediction ability, especially in the late software testing phase.

In both time-domain and group data sets, when we compare the PMSE between the best Burr-type NHPP-based SRM and the best SRATS SRM, we find out that our Burr-type NHPP-based SRMs could guarantee smaller PMSEs than the SRATS SRMs in many cases; 3 cases in (i), half of the cases in (ii), and 6 cases in (iii) of Table 9, and half of the data sets in (i), 7 out of 8 data sets in (ii) and 6 out of 8 sets of Table 10.

We never claim here that the Burr-type NHPP-based SRMs are always better than the existing SRMs in the literature. However,

Table 9
Predictive performances based on AIC (time-domain data).

Data Set	Burr Type			SRATS		
	Best Burr	AIC	PMSE	Best SRATS	AIC	PMSE
(i) Prediction from the 20% observation point						
TDS1	Burr-type X	141.639	5252.410	exp	**141.609**	**34.435**
TDS2	Burr-type XII	91.195	270.380	tnorm	**84.722**	**42.104**
TDS3	Log Burr-type IX	**313.458**	**885.923**	llogist	313.745	1059.240
TDS4	Log Burr-type IX	126.424	**64.510**	exp	**121.858**	387.312
TDS5	Tru Burr-type IX	113.664	37162.600	exp	**113.372**	**9711.670**
TDS6	Tru Burr-type VIII	128.996	273.412	lxvmax	**128.656**	**167.226**
TDS7	Tru Burr-type IX	189.504	89821.400	exp	**187.583**	**794.703**
TDS8	Tru Burr-type IX	440.966	6.355E+05	exp	**440.510**	**1.399E+05**
(ii) Prediction from the 50% observation point						
TDS1	Burr-type X	403.494	167.253	exp	**403.368**	**103.723**
TDS2	Log Burr-type VIII	256.977	219.776	exp	**256.074**	**14.890**
TDS3	Log Burr-type IX	**861.677**	1092.960	llogist	861.949	**960.575**
TDS4	Log Burr-type IX	**334.737**	580.186	exp	334.762	**106.263**
TDS5	Log Burr-type IX	364.639	**310.799**	exp	**363.831**	381.855
TDS6	Log Burr-type VIII	344.810	**4.358**	lxvmax	**344.604**	10.493
TDS7	Tru Burr-type IX	**445.193**	**37.702**	tlogist	445.247	3.924E+06
TDS8	Burr-type X	1094.480	**1.651E+05**	exp	**1092.710**	2.349E+05
(iii) Prediction from the 80% observation point						
TDS1	Log Burr-type VIII	**691.675**	**5.720**	lxvmax	691.677	5.772
TDS2	Log Burr-type VIII	443.895	**0.583**	lxvmax	**443.891**	0.588
TDS3	Log Burr-type IX	**1478.400**	**25.476**	llogist	1478.500	26.709
TDS4	Log Burr-type X	566.040	**2.867**	exp	**565.497**	4.253
TDS5	Log Burr-type IX	**577.096**	62.066	llogist	577.358	**51.671**
TDS6	Log Burr-type VII	553.819	5.711	lxvmax	**553.472**	**2.041**
TDS7	Tru Burr-type IX	**769.494**	**16.783**	exp	769.836	20.733
TDS8	Tru Burr-type IX	**1886.600**	**130.770**	pareto	1889.040	148.226

Table 10
Predictive performances based on AIC (group data).

Data Set	Burr Type			SRATS		
	Best Burr	AIC	PMSE	Best SRATS	AIC	PMSE
(i) Prediction from the 20% observation point						
GDS1	Tru Burr-type IX	12.704	**225.282**	exp	**11.085**	625.983
GDS2	Log Burr-type IX	12.865	**19.943**	lxvmax	12.865	29.244
GDS3	Log Burr-type IX	18.976	**429.795**	exp	**18.442**	1860.480
GDS4	Tru Burr-type VIII	12.649	798.003	exp	**11.669**	**142.854**
GDS5	Log Burr-type VII	8.000	3.535	exp	**7.386**	**2.628**
GDS6	Tru Burr-type VIII	20.698	490.855	lnorm	**20.660**	**478.505**
GDS7	Log Burr-type III	17.340	**1251.860**	txvmin	**16.958**	1264.850
GDS8	Burr-type X	8.614	1799.630	txvmin	8.614	**423.360**
(ii) Prediction from the 50% observation point						
GDS1	Tru Burr-type IX	**34.496**	**38.234**	lxvmin	36.846	209.864
GDS2	Log Burr-type IX	**29.849**	**48.829**	lxvmax	31.051	63.385
GDS3	Log Burr-type VI	**48.466**	**555.562**	exp	49.313	1050.130
GDS4	Tru Burr-type IX	**30.493**	**415.168**	tlogist	30.560	4310.030
GDS5	Log Burr-type IX	17.787	**0.300**	exp	**17.365**	0.344
GDS6	Log Burr-type IX	**38.584**	**327.310**	lxvmax	40.521	384.078
GDS7	Log Burr-type IX	**71.671**	**22.810**	lxvmax	72.390	22.894
GDS8	Tru Burr-type VI	**63.463**	46.071	txvmin	65.835	**29.110**
(iii) Prediction from the 80% observation point						
GDS1	Tru Burr-type IX	**55.028**	**12.006**	lxvmin	56.861	12.076
GDS2	Burr-type IX	**50.295**	**0.836**	lxvmax	52.523	0.927
GDS3	Log Burr-type VI	**73.981**	**3.123**	txvmin	75.292	4.917
GDS4	Log Burr-type VI	43.129	**1.371**	txvmin	**42.540**	3.656
GDS5	Log Burr-type IX	24.517	1.998	exp	**24.271**	**0.436**
GDS6	Burr-type XII	**93.323**	**1.710**	lxvmax	96.179	5.570
GDS7	Log Burr-type IX	**112.543**	118.689	txvmin	112.836	**6.118**
GDS8	Tru Burr-type IX	**100.305**	**5.942**	tlogist	100.325	5.970

we emphasize that the Burr-type NHPP-based SRMs should be the possible candidates in selecting the best SRM in terms of goodness-of-fit and predictive performances. Also, another new finding is that the logarithmic and truncated Burr-type NHPP-based SRMs gave better goodness-of-fit and prediction results in many cases than the existing Burr-type III, X, and XII SRMs. This would be useful to assume the competitors in SRMs.

4.4. Software reliability assessment

Finally, we evaluate the software reliability quantitatively with our Burr-type NHPP-based SRMs and compare them with the existing NHPP-based SRMs in SRATS. Let $R(x \mid t)$ be the software reliability with the software operational period (prediction length) $x = t_{m_T+l} - t_{m_T}$ or l when the software is released at time

Table 11
Software reliability assessment with the best AIC (time-domain data).

Burr Type			SRATS	
	Best Burr	Reliability	Best SRATS	Reliability
TDS1	Log Burr-type VIII	2.631E−06	lxvmax	**2.674E−06**
TDS2	Log Burr-type VIII	3.687E−03	lxvmax	**3.751E−03**
TDS3	Burr-type X	**4.592E−05**	lxvmin	2.516E−10
TDS4	Tru Burr-type IX	1.244E−01	pareto	**1.000E−00**
TDS5	Burr-type X	**1.035E−05**	exp	2.596E−08
TDS6	Log Burr-type VIII	3.283E−04	lxvmax	**4.694E−03**
TDS7	Log Burr-type VII	**2.453E−04**	lxvmax	2.398E−04
TDS8	Tru Burr-type VI	6.158E−10	pareto	**7.736E−06**

Table 12
Software reliability assessment with the best AIC (group data).

Burr Type			SRATS	
	Best Burr	Reliability	Best SRATS	Reliability
GDS1	Log Burr-type IX	**1.065E−02**	llogist	4.152E−03
GDS2	Log Burr-type IX	1.353E−05	lxvmax	**7.236E−05**
GDS3	Log Burr-type VI	3.751E−02	tnorm	**3.865E−02**
GDS4	Tru Burr-type IX	**4.504E−01**	tlogist	2.816E−01
GDS5	Log Burr-type IX	**6.548E−03**	exp	9.832E−04
GDS6	Log Burr-type IX	1.928E−08	lxvmax	**1.939E−07**
GDS7	Burr-type III	8.667E−01	txvmin	**9.633E−01**
GDS8	Log Burr-type IX	**6.679E−01**	llogist	6.373E−01

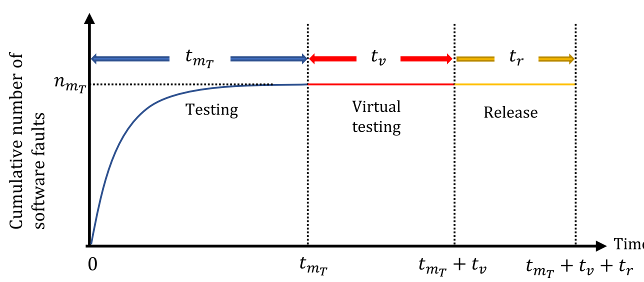


Fig. 4. Software release decision based on virtual testing time.

$t = t_{m_G}$. Since $R(x \mid t)$ is defined as the probability that software is fault-free during the time interval $(t, t+x]$, it is easily obtained that

$$R(x \mid t) = \Pr(N(t+x) - N(t) = 0 \mid N(t) = m_T)$$
$$= \exp\left(-\left[\Lambda(t+x;\boldsymbol{\theta}) - \Lambda(t;\boldsymbol{\theta})\right]\right), \quad (20)$$

where m_T is the cumulative number of software faults detected up to time t in the time-domain data (m_T in Eq. (20) is replaced by n_{m_G} in the group data). In our subsequent examples, we suppose that the prediction length x is equivalent to the testing length experienced before, say, $t = x$.

Tables 11 and 12 present the quantitative software reliability. We assume the Burr-type NHPP-based SRM and the SRATS NHPP-based SRM with the minimum AIC in the fault-detection time-domain and group data sets, where the bold font denotes the case with a greater reliability estimate. Looking at these results, it is seen that our Burr-type NHPP-based SRMs could show larger software reliability estimates than the existing NHPP-based SRMs in 3 out of 8 cases (time-domain data) and 4 out of 8 cases (group data). This feature tells us that the Burr-type NHPP-based SRMs tend to make more optimistic decisions in software reliability assessment than the SRATS NHPP-based SRMs. It is worth noting in all the data sets that after each observation point, software faults were additionally detected as the ex-post results. Hence, the optimistic reliability estimation is not preferable. Fig. 5(a) and (b) show the software reliability estimates with the Burr-type NHPP-based SRM and the SRATS NHPP-based SRM in TDS1 and GDS1, respectively. In both cases, the software reliability values dropped down to zero level rapidly, but two NHPP-based SRMs showed similar reliability values as well. From these results, we find out that both SRMs gave the false alarm to release the current software at respective observation points and requested more testing to attain the requirement level of software reliability.

Next, we introduce the concept of *virtual testing time*, whose idea comes from Zhao et al. (2018) to consider a more realistic and plausible software release decision. When the software testing is terminated at a given observed time point, we set the so-called virtual testing time period when no software fault is found. If this hypothesis is correct, we check whether the software reliability can achieve a given requirement level at the end of the virtual testing time and release the software product with a satisfactory level at the end of the virtual testing time period. Otherwise, *i.e.*, if at least one software fault was found, we reset the observation point to the fault detection/fixing time and redefine the virtual testing time from that point. Under the hypothesis that no fault is found during the virtual testing time, the maximum likelihood estimation is made with zero fault count. In the time-domain data, the likelihood function in Eq. (6) is given by

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{T}) = \exp(-\Lambda(t_{m_T}+t_v;\boldsymbol{\theta}))\prod_{i=1}^{m_T}\lambda(t_i;\boldsymbol{\theta}), \quad (21)$$

where t_v is the virtual testing time (see Fig. 4). In the group data, the likelihood function in Eq. (8) is given by

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{I}) = \prod_{i=1}^{m_G+v}\left[\frac{[\Lambda(t_i;\boldsymbol{\theta}) - \Lambda(t_{i-1};\boldsymbol{\theta})]^{n_i-n_{i-1}}}{(n_i-n_{i-1})!}\right]$$
$$\times e^{-[\Lambda(t_i;\boldsymbol{\theta})-\Lambda(t_{i-1};\boldsymbol{\theta})]}, \quad (22)$$

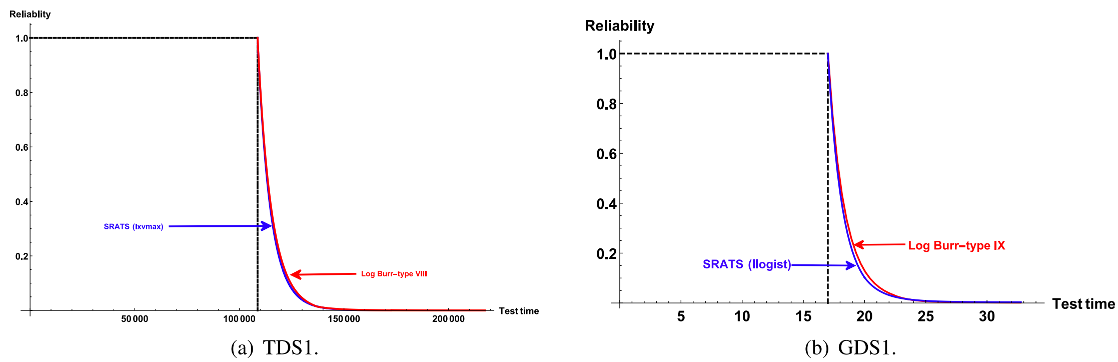


Fig. 5. Predictive software reliability assessment with the best Burr-type and SRATS NHPP-based SRMs.

where $(t_i, n_i) = (t_{m_i}, n_{m_G})$ $(i = m_G + 1, m_G + 2, \ldots, m_G + v)$ with the virtual testing time length v (integer value). Hence, it is obvious that the maximum likelihood estimates depend on the length of virtual testing time t_v or v so that increasing virtual testing time leads to increasing quantitative software reliability. Then the problem is to determine an appropriate virtual testing time (t_v^* or v^*) satisfying that the software reliability with a given operational period is greater than a specified requirement lever, *e.g.*, such as 90%.

In our numerical experiments, we focus on the time-domain and group data and set 15 different lengths of virtual testing time (10% to 150% of t_{m_T} (t_{m_G})), where the operational period x is given by $x = t_r = t_{m_T}$ or t_{m_G} and each t_{m_T} or t_{m_G} is given in Table 4 (a) and (b). Tables 13 and 14 present the software reliability prediction with the Burr-type NHPP-based SRM and SRATS NHPP-based SRM with the minimum AIC in time-domain data and group data, when the virtual testing time is given by 10% to 150% length of the testing time t_{m_T} or t_{m_G}. In almost all cases, it is seen that the longer the virtual testing period, the closer the software reliability value to unity. Based on the assumption that no software fault is found during the virtual testing time, NHPP-based SRMs could provide much higher software reliability estimates than the results in Tables 11 and 12. In other words, it is impossible to guarantee a satisfactory software reliability estimate without setting up the virtual testing time long enough, which implies the belief that the software product is reliable. In Tables 13 and 14, we seek the virtual testing time when the specified reliability level is given by 90%. For instance, in GDS1, we find that the virtual testing time when the reliability is greater than 90% becomes 130% and 180% of the testing time for the Burr-type NHPP-based SRM and the SRATS NHPP-base SRM, respectively, so that longer virtual testing time with zero fault count than the testing length is required to achieve the requirement because the quantitative software itself is the belief by the tester. In some cases, it is seen that 90% software reliability requirement seems to be unrealistic because the virtual testing time with zero fault count must be dozens of times t_{m_T} or t_{m_G}, *i.e.*, in TDS1, TDS2, TDS6, TDS7, and GDS6. On the one hand, in TDS4 and GDS7, the 90% requirement level is achieved after the software testing when SRATS NHPP-based SRM is used. Of course, the software release decision considered here is based on the existence of virtual testing time with zero fault count. If any software fault was detected during the period, the observation point to trigger the virtual testing is changed step by step.

5. Related work

Burr (1942) proposed an interesting family of continuous probability distributions, including 12 types of CDFs (I $\sim$ XII), which yield various probability density shapes. Since the Burr-type distributions have monotone and/or unimodal failure rates, some of them have been often used for lifetime data analysis (Tadikamalla, 1980; Zimmer et al., 1998). Abdel-Ghaly et al. (1997) applied the Burr-type XII distribution to software reliability growth modeling for the first time, where they concerned the generalized order statistics SRM (Miller, 1986) and the Bayesian inference, and further considered an NHPP-based SRM with the Burr-type XII fault-detection time distribution for a one-stage look ahead prediction with the fault-detection time-domain data. Kim and Park (2010) and Kim (2013) also assumed the Burr-type XII fault-detection time distribution in the NHPP-based SRM and applied it to the optimal software release problem and statistical process control chart, respectively. Ann (An, 2012) treated both the order statistics-based and NHPP-based SRMs with the Burr-type XII distribution with the fault-detection time-domain data and compared them in terms of the data fitting by U-plot and Y-plot using Kolmogorov distance (Lyu, 1996).

Prasad et al. (2014b) also conducted the maximum likelihood estimation for the NHPP-based SRM with the Burr-type XII distribution and analyzed 5 reference data sets with the fault-detection time-domain data. The same authors (Prasad et al., 2014c) sequentially predicted the number of software fault count with the same model and examined a statistical process control chart in a fashion similar to Kim (2013). Prasad et al. (2014a) analyzed the fault-detection time-interval data (group data) with the Burr-type XII NHPP-based SRM and constructed a statistical process control chart. Ravikumar and Kantam (2017) also estimated the model parameters in the NHPP-based SRM with the Burr-type XII with the group data by means of the least-squares estimation. Islam (2020) assumed the Burr-type XII testing-effort function for the NHPP-based SRM with a trend-change point.

Ahmad et al. (2009, 2011) assumed a different type of Burr-type distribution, say, the Burr-type III distribution, to describe the testing effort for software fault count processes in the NHPP-based software reliability modeling, and applied it to the software release problems. Sobhana and Prasad (2015) and Chowdary et al. (2015) used the Burr-type III distribution for the generalized order statistics SRM and the NHPP-based SRM, respectively, where the fault-detection time-domain data were analyzed. Sridevi and Rani (2015) compared two baseline models with the Burr-type XII and the Burr-type III distributions in the NHPP-based modeling framework with the fault-detection time-domain data. Yet another Burr-type X distribution was introduced by Sridevi and Akbar (2016) to propose a different NHPP-based SRM, where the same fault-detection time-domain data (Prasad et al., 2014b; Chowdary et al., 2015) were analyzed. Recently, Kim (2020) introduced the Burr-Hatke-exponential distribution in the NHPP-based software reliability modeling and compared it with the common exponential distribution (Goel and Okumoto, 1979) and the inverse exponential distribution with the fault-detection time-domain data.

Table 13
Reliability prediction with virtual testing time in time-domain data.

Virtual testing time	TDS1 Burr Type (Log Burr-type VIII)	TDS1 SRATS (lxvmax)	TDS2 Burr Type (Log Burr-type VIII)	TDS2 SRATS (lxvmax)	TDS3 Burr Type (Burr-type X)	TDS3 SRATS (lxvmin)	TDS4 Burr Type (Tru Burr-type IX)	TDS4 SRATS (pareto)
10%	2.648E−05	2.778E−05	8.831E−03	9.117E−03	6.174E−02	1.110E−06	4.284E−01	**1.000E+00**
20%	1.434E−04	1.518E−04	1.728E−02	1.787E−02	3.267E−01	1.512E−04	**9.717E−01**	1.000E+00
30%	5.167E−04	5.465E−04	2.924E−02	3.022E−02	6.260E−01	2.981E−03	1.000E+00	1.000E+00
40%	1.407E−03	1.480E−03	4.458E−02	4.597E−02	8.209E−01	1.933E−02	1.000E+00	1.000E+00
50%	3.189E−03	1.847E−05	6.293E−02	6.472E−02	**9.213E−01**	6.467E−02	1.000E+00	1.000E+00
60%	6.049E−03	6.244E−03	8.371E−02	8.591E−02	9.675E−01	8.591E−02	1.000E+00	1.000E+00
70%	1.050E−02	1.064E−02	1.068E−01	1.090E−01	9.872E−01	2.486E−01	1.000E+00	1.000E+00
80%	1.616E−02	1.664E−02	1.305E−01	1.334E−01	9.953E−01	3.626E−01	1.000E+00	1.000E+00
90%	2.371E−02	2.431E−02	1.555E−01	1.587E−01	9.983E−01	4.733E−01	1.000E+00	1.000E+00
100%	3.291E−02	3.365E−02	1.811E−01	1.844E−01	9.994E−01	5.726E−01	1.000E+00	1.000E+00
110%	4.370E−02	4.455E−02	2.067E−01	2.102E−01	9.998E−01	6.575E−01	1.000E+00	1.000E+00
120%	5.594E−02	5.694E−02	2.323E−01	2.359E−01	9.999E−01	7.278E−01	1.000E+00	1.000E+00
130%	6.950E−02	7.060E−02	2.571E−01	2.612E−01	1.000E+00	7.847E−01	1.000E+00	1.000E+00
140%	8.418E−02	8.536E−02	2.822E−01	2.859E−01	1.000E+00	8.302E−01	1.000E+00	1.000E+00
150%	9.981E−02	1.011E−01	3.064E−01	3.101E−01	1.000E+00	8.663E−01	1.000E+00	1.000E+00
Virtual test time required to reach 90% reliability	1550%	1550%	1110%	1100%	50%	270%	20%	0%

Virtual testing time	TDS5 Burr Type (Burr-type X)	TDS5 SRATS (exp)	TDS6 Burr Type (Log Burr-type VIII)	TDS6 SRATS (lxvmax)	TDS7 Burr Type (Log Burr-type VII)	TDS7 SRATS (lxvmax)	TDS8 Burr Type (Tru Burr-type VI)	TDS8 SRATS (pareto)
10%	2.759E−02	2.892E−05	1.597E−03	1.566E−02	1.157E−03	1.144E−03	3.153E−04	6.944E−05
20%	2.505E−01	1.285E−03	2.433E−03	3.586E−02	3.657E−03	3.534E−03	8.535E−04	3.586E−04
30%	5.606E−01	1.197E−02	5.290E−03	6.520E−02	8.312E−03	8.267E−03	4.304E−03	1.256E−03
40%	7.797E−01	4.812E−02	1.494E−02	1.020E−01	1.595E−02	1.591E−02	7.400E−01	3.381E−03
50%	8.984E−01	1.190E−01	2.660E−02	1.439E−01	2.686E−02	2.683E−02	**9.765E−01**	4.069E−03
60%	**9.557E−01**	2.187E−01	3.086E−02	1.888E−01	4.097E−02	4.096E−02	9.975E−01	1.404E−02
70%	9.815E−01	3.326E−01	5.698E−02	2.348E−01	5.802E−02	5.801E−02	9.999E−01	2.369E−02
80%	9.927E−01	4.466E−01	6.776E−02	2.803E−01	7.760E−02	7.757E−02	1.000E+00	3.659E−02
90%	9.972E−01	5.513E−01	7.570E−02	3.246E−01	9.921E−02	9.925E−02	1.000E+00	5.270E−02
100%	9.990E−01	6.420E−01	1.299E−01	3.671E−01	1.224E−01	1.224E−01	1.000E+00	7.189E−02
110%	9.997E−01	7.176E−01	1.508E−01	4.072E−01	1.465E−01	1.467E−01	1.000E+00	9.349E−02
120%	9.999E−01	7.790E−01	1.710E−01	4.450E−01	1.664E−01	1.718E−01	1.000E+00	1.173E−01
130%	1.000E+00	8.281E−01	2.021E−01	4.802E−01	1.896E−01	1.971E−01	1.000E+00	1.429E−01
140%	1.000E+00	8.667E−01	2.312E−01	5.130E−01	2.157E−01	2.226E−01	1.000E+00	1.696E−01
150%	1.000E+00	8.970E−01	2.331E−01	5.435E−01	2.328E−01	2.478E−01	1.000E+00	1.972E−01
Virtual test time required to reach 90% reliability	60%	160%	1240%	560%	1390%	1070%	50%	880%

The above references mentioned that different Burr-type distributions are introduced in different model settings (generalized order statistics SRM and NHPP-based SRM) and different software fault count data types (time-domain data and group data). Unfortunately, no comprehensive comparison with the existing SRMs was made with different fault count data types. Imanaka and Dohi (2015) compared the NHPP-based SRM under the Burr-type XII distribution with the representative 11 NHPP-based SRMs (Goel and Okumoto, 1979; Yamada et al., 1983; Zhao and Xie, 1996; Ohba, 1984; Gokhale and Trivedi, 1998; Abdel-Ghaly et al., 1986; Okamura et al., 2013; Achcar et al., 1998; Ohishi et al., 2009) with 8 software fault-count group data, and further proposed the Burr-type XII regression SRM modulated by an NHPP when software process metrics data are given. They concluded that the Burr-type distribution is quite attractive to represent the software fault-detection time distribution because the goodness-of-fit performances for the NHPP-based SRM with Burr-type XII distribution were better in many cases in terms of the Akaike information criterion (AIC) and mean squares error (MSE). However, the Ravikumar and Kantam (2017) did not investigate the other Burr-type distributions and the predictive performances in the future testing period. In this paper, we developed 11 Burr-type NHPP-based SRMs, including the Burr-type III, X, and XII distributions, and compared them with the well-known NHPP-based SRMs under the time-domain data and group data circumstances. This is the most comprehensive study to evaluate the Burr-type NHPP-based SRMs and provides the empirical basis for why the

Table 14
Reliability prediction with virtual testing time in group data.

	GDS1		GDS2		GDS3		GDS4	
Virtual testing time	Burr Type (Log Burr-type IX)	SRATS (llogist)	Burr Type (Log Burr-type IX)	SRATS (lxvmax)	Burr Type (Log Burr-type VI)	SRATS (tnorm)	Burr Type (Tru Burr-type IX)	SRATS (tlogist)
10%	3.846E−02	1.749E−02	5.844E−05	3.721E−04	1.433E−01	1.918E−01	7.148E−01	5.470E−01
20%	1.518E−01	8.713E−02	1.588E−04	1.232E−03	5.084E−01	6.276E−01	8.621E−01	7.384E−01
30%	3.073E−01	2.003E−01	7.201E−04	6.245E−03	7.662E−01	8.738E−01	**9.159E−01**	8.543E−01
40%	3.785E−01	2.630E−01	1.232E−03	1.106E−02	**9.139E−01**	**9.629E−01**	9.685E−01	**9.200E−01**
50%	5.137E−01	3.849E−01	6.354E−03	2.600E−02	9.596E−01	9.901E−01	9.919E−01	9.761E−01
60%	6.215E−01	4.926E−01	1.315E−02	3.605E−02	9.830E−01	9.976E−01	9.970E−01	9.869E−01
70%	6.669E−01	5.331E−01	2.322E−02	4.763E−02	9.939E−01	9.995E−01	9.986E−01	9.928E−01
80%	7.055E−01	6.208E−01	5.250E−02	7.499E−02	9.970E−01	9.999E−01	9.994E−01	9.961E−01
90%	7.877E−01	6.864E−01	7.220E−02	8.968E−02	9.991E−01	1.000E+00	9.997E−01	9.978E−01
100%	8.268E−01	7.390E−01	1.166E−01	1.214E−01	9.996E−01	1.000E+00	1.000E+00	9.994E−01
110%	8.496E−01	7.613E−01	1.408E−01	1.379E−01	9.998E−01	1.000E+00	1.000E+00	9.996E−01
120%	8.710E−01	7.994E−01	1.658E−01	1.546E−01	9.999E−01	1.000E+00	1.000E+00	9.998E−01
130%	**9.002E−01**	8.301E−01	2.170E−01	1.884E−01	1.000E+00	1.000E+00	1.000E+00	9.999E−01
140%	9.151E−01	8.433E−01	2.425E−01	2.051E−01	1.000E+00	1.000E+00	1.000E+00	9.999E−01
150%	9.241E−01	8.660E−01	2.926E−01	2.381E−01	1.000E+00	1.000E+00	1.000E+00	1.000E+00
Virtual test time required to reach 90% reliability	130%	180%	350%	1470%	40%	40%	30%	40%

	GDS5		GDS6		GDS7		GDS8	
Virtual testing time	Burr Type (Log Burr-type IX)	SRATS (exp)	Burr Type (Log Burr-type IX)	SRATS (lxvmax)	Burr Type (Burr-type III)	SRATS (txvmin)	Burr Type (Log Burr-type IX)	SRATS (llogist)
10%	3.595E−02	8.250E−03	2.321E−07	4.471E−06	**9.638E−01**	**9.993E−01**	7.970E−01	7.870E−01
20%	2.167E−01	2.717E−02	1.597E−06	3.786E−05	9.902E−01	1.000E+00	8.719E−01	8.752E−01
30%	5.625E−01	5.967E−02	6.094E−06	1.779E−04	9.970E−01	1.000E+00	**9.370E−01**	**9.204E−01**
40%	4.455E−01	7.138E−02	2.256E−05	5.724E−04	9.993E−01	1.000E+00	9.560E−01	9.489E−01
50%	5.956E−01	8.958E−02	4.912E−05	1.419E−03	9.997E−01	1.000E+00	9.601E−01	9.662E−01
60%	6.607E−01	9.725E−02	1.774E−04	2.928E−03	9.999E−01	1.000E+00	9.791E−01	9.770E−01
70%	6.834E−01	1.045E−01	6.631E−04	5.287E−03	1.000E+00	1.000E+00	9.842E−01	9.840E−01
80%	7.766E−01	1.185E−01	1.811E−03	8.631E−03	1.000E+00	1.000E+00	9.910E−01	9.887E−01
90%	8.494E−01	1.255E−01	3.987E−03	1.304E−02	1.000E+00	1.000E+00	9.955E−01	9.918E−01
100%	8.869E−01	1.401E−01	7.506E−03	1.852E−02	1.000E+00	1.000E+00	9.940E−01	9.940E−01
110%	8.893E−01	1.478E−01	1.260E−02	2.505E−02	1.000E+00	1.000E+00	9.977E−01	9.955E−01
120%	8.957E−01	1.558E−01	1.938E−02	3.257E−02	1.000E+00	1.000E+00	9.983E−01	9.966E−01
130%	**9.298E−01**	1.727E−01	2.785E−02	4.101E−02	1.000E+00	1.000E+00	9.988E−01	9.974E−01
140%	9.309E−01	1.817E−01	3.793E−02	5.023E−02	1.000E+00	1.000E+00	9.991E−01	9.980E−01
150%	9.467E−01	2.007E−01	3.795E−02	6.019E−02	1.000E+00	1.000E+00	9.980E−01	9.980E−01
Virtual test time required to reach 90% reliability	130%	750%	1890%	3380%	10%	0%	30%	30%

Burr-type distributions are appropriate to describe the software fault-detection time distribution.

6. Conclusions

In this paper, we have developed the Burr-type NHPP-based SRMs and compared them with the existing SRMs in the past literature in terms of goodness-of-fit and predictive performances. Throughout numerical experiments with 8 fault-detection time-domain data sets and 8 group data sets, which were observed in actual software development projects, we have confirmed that our Burr-type NHPP-based SRMs could show the better performances in many cases than the existing 11 NHPP-based SRMs in SRATS. The lessons learned from our numerical examples in the following:

(i) Burr-type NHPP-based SRMs could provide the better goodness-of-fit performances than the existing NHPP-based SRMs in SRATS in 11 out of 16 data sets.

(ii) Based on PMSE, our Burr-type NHPP-based SRMs had the better potential for accurate prediction of unknown future fault detection than the existing NHPP-based SRMs in the half of group data sets.

(iii) In three observation points of group data sets, our Burr-type NHPP-based SRMs were superior to the existing NHPP-based SRMs in terms of the predictive performance in many cases on the scenario that the best model is selected in terms of the minimum AIC.

The main contribution of the paper consists in suggesting that the Burr-type NHPP-based SRMs are quite attractive SRMs to describe the software fault-detection processes and should be the possible candidates in selecting the best SRM in terms of goodness-of-fit and predictive performances. This fact has not been known during the last four decades.

In the future, it is beneficial to implement the Burr-type NHPP-based SRMs on the well-established software reliability assessment tool. Although SRATS (Okamura and Dohi, 2013) contains 11 well-known NHPP-based SRMs, the main feature is to guarantee the global convergence of model parameters in computing the ML estimates, where the EM (Expectation–Maximization) algorithms are implemented for the respective SRMs. In order to implement the reliable and automated ML prediction for the Burr-type NHPP-based SRMs, we need to design the EM algorithms for our 11 Burr-type NHPP-based SRMs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdel-Ghaly, A.A., Al-Dayian, G.R., Al-Kashkari, F.H., 1997. The use of Burr type XII distribution on software reliability growth modelling. Microelectr. Reliab. 37 (2), 305–313. http://dx.doi.org/10.1016/0026-2714(95)00124-7.

Abdel-Ghaly, A.A., Chan, P.Y., Littlewood, B., 1986. Evaluation of competing software reliability predictions. IEEE Trans. Softw. Eng. SE-12 (9), 950–967. http://dx.doi.org/10.1109/TSE.1986.6313050.

Achcar, J.A., Dey, D.K., Niverthi, M., 1998. A Bayesian approach using nonhomogeneous Poisson processes for software reliability models. In: Front. Reliab.. World Scientific, pp. 1–18. http://dx.doi.org/10.1142/9789812816580_0001.

Ahmad, N., Khan, M.G.M., Quadri, S.M.K., Kumar, M., 2009. Modelling and analysis of software reliability with Burr type X testing-effort and release-time determination. J. Modell. Manag. 4 (1), 28–54. http://dx.doi.org/10.1108/17465660910943748.

Ahmad, N., Quadri, S.M.K., Khan, M.G.M., Kumar, M., 2011. Software reliability growth models incorporating Burr type III test-effort and cost-reliability analysis. Int. J. Comput. Sci. Inf. Technol. 2 (1), 555–562.

An, J.-H., 2012. Two model comparisons of software reliability analysis for Burr type XII distribution. J. Korean Data Inf. Sci. Soc. 23 (4), 815–823. http://dx.doi.org/10.7465/jkdi.2012.23.4.815.

Burr, I.W., 1942. Cumulative frequency functions. Ann. Math. Stat. 13 (2), 215–232. http://dx.doi.org/10.1214/aoms/1177731607.

Chen, Y., Singpurwalla, N.D., 1997. Unification of software reliability models by self-exciting point processes. Adv. Appl. Probab. 29 (2), 337–352. http://dx.doi.org/10.2307/1428006.

Chowdary, C.S., Prasad, R.S., Sobhana, K., 2015. Burr type III software reliability growth model. IOSR J. Comput. Eng. 17 (1), 49–54. http://dx.doi.org/10.9790/0661-17144954.

Goel, A.L., 1985. Software reliability models: assumptions, limitations, and applicability. IEEE Trans. Softw. Eng. SE-11 (12), 1411–1423. http://dx.doi.org/10.1109/TSE.1985.232177.

Goel, A.L., Okumoto, K., 1979. Time-dependent error-detection rate model for software reliability and other performance measures. IEEE Trans. Reliab. R-28, 206–211. http://dx.doi.org/10.1109/TR.1979.5220566.

Gokhale, S.S., Trivedi, K.S., 1998. Log-logistic software reliability growth model. In: Proceedings Third IEEE International High-Assurance Systems Engineering Symposium. HASE 1998, pp. 34–41. http://dx.doi.org/10.1109/HASE.1998.731593.

Gokhale, S.S., Trivedi, K.S., 1999. A time/structure based software reliability model. Ann. Softw. Eng. 8 (1), 85–121. http://dx.doi.org/10.1023/A:1018923329647.

Huang, C.-Y., Lyu, M.R., Kuo, S.-Y., 2003. A unified scheme of some nonhomogenous Poisson process models for software reliability estimation. IEEE Trans. Softw. Eng. 29 (3), 261–269. http://dx.doi.org/10.1109/TSE.2003.1183936.

Imanaka, T., Dohi, T., 2015. Software reliability modeling based on Burr XII distributions. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 98 (10), 2091–2095. http://dx.doi.org/10.1587/transfun.E98.A.2091.

Islam, S.F., 2020. Introducing Burr-type XII testing-effort with change point based software reliability growth model. Glob. Sci. J. 8 (8), 2712–2718.

Kim, H.-C., 2013. Assessing software reliability based on NHPP using SPC. Int. J. Softw. Eng. Appl. 7 (6), 61–70. http://dx.doi.org/10.14257/ijseia.2013.7.6.06.

Kim, H.-C., 2020. A study on comparative evaluation of software reliability model applying modified exponential distribution. Int. J. Eng. Res. Technol. 13 (5), 867–872. http://dx.doi.org/10.37624/ijert/13.5.2020.867-872.

Kim, H.-C., Park, H.-K., 2010. The comparative study of software optimal release time based on Burr distribution. Int. J. Adv. Comput. Technol. 2 (3), 119–128. http://dx.doi.org/10.4156/ijact.vol2.issue3.13.

Langberg, N., Singpurwalla, N.D., 1985. A unification of some software reliability models. SIAM J. Sci. Stat. Comput. 6 (3), 781–790. http://dx.doi.org/10.1137/0906053.

Li, S., Dohi, T., Okamura, H., 2021. A comprehensive evaluation for Burr-type NHPP-based software reliability models. In: 2021 8th International Conference on Dependable Systems and their Applications. DSA 2021, pp. 1–12.

Lyu, M.R. (Ed.), 1996. Handbook of Software Reliability Engineering. McGraw-Hill, NewYork.

Miller, D.R., 1986. Exponential order statistic models of software reliability growth. IEEE Trans. Softw. Eng. SE-12 (1), 12–24. http://dx.doi.org/10.1109/TSE.1986.6312915.

Musa, J.D., 1979. Software Reliability Data. Technical Report in Rome Air Development Center.

Musa, J.D., Iannino, A., Okumoto, K., 1987. Software Reliability - Measurement, Prediction, Application. McGraw-Hill, NewYork.

Ohba, M., 1984. Inflection S-shaped software reliability growth model. In: Stochastic Models in Reliability Theory. Springer, pp. 144–162. http://dx.doi.org/10.1007/978-3-642-45587-2_10.

Ohishi, K., Okamura, H., Dohi, T., 2009. Gompertz software reliability model: estimation algorithm and empirical validation. J. Syst. Softw. 82 (3), 535–543. http://dx.doi.org/10.1016/j.jss.2008.11.840.

Okamura, H., Dohi, T., 2013. SRATS: software reliability assessment tool on spreadsheet (experience report). In: 2013 IEEE 24th International Symposium on Software Reliability Engineering. ISSRE 2013, pp. 100–107. http://dx.doi.org/10.1109/ISSRE.2013.6698909.

Okamura, H., Dohi, T., 2016. Phase-type software reliability model: parameter estimation algorithms with grouped data. Ann. Oper. Res. 244 (1), 177–208. http://dx.doi.org/10.1007/s10479-015-1870-0.

Okamura, H., Dohi, T., Osaki, S., 2013. Software reliability growth models with normal failure time distributions. Reliab. Eng. Syst. Saf. 116, 135–141. http://dx.doi.org/10.1016/j.ress.2012.02.002.

Okamura, H., Etani, Y., Dohi, T., 2011. Quantifying the effectiveness of testing efforts on software fault detection with a logit software reliability growth model. In: 2011 Joint Conference of the 21st International Workshop on Software Measurement (IWSM 2011) and the 6th International Conference on Software Process and Product Measurement (MENSURA 2011). pp. 62–68. http://dx.doi.org/10.1109/IWSM-MENSURA.2011.26.

Prasad, R.S., Mohan, K.V.M., Sridevi, G., 2014a. Assessing Burr-type XII software reliability for interval domain data using SPC. Comput. Eng. 79, 30335–30340.

Prasad, R.S., Mohan, K.M., Sridevi, G., 2014b. Burr type XII software reliability growth model. Int. J. Comput. Appl. 108 (16), 16–20. http://dx.doi.org/10.5120/18995-0452.

Prasad, R.S., Mohan, K.V.M., Sridevi, G., 2014c. Monitoring Burr type XII software quality using SPC. Int. J. Appl. Eng. Res. 9 (22), 16651–16660.

Ravikumar, M.S., Kantam, R.R.L., 2017. Software reliability model based on Burr-type XII distribution. Int. J. Adv. Eng. Res. Appl. 2 (9), 561–564.

Sobhana, K., Prasad, R.S., 2015. Burr-type III software reliability with SPC - an order statistics approach. Int. J. Res. Stud. Comput. Sci. Eng. 2 (3), 21–24.

Sridevi, G., Akbar, S., 2016. Burr-type X software reliability growth model. Asian J. Inf. Technol. 15 (16), 2988–2991. http://dx.doi.org/10.36478/ajit.2016.2988.2991.

Sridevi, G., Rani, C.M.S., 2015. Comparison of software reliability analysis for Burr distribution. J. Theor. Appl. Inf. Technol. 81 (1), 144–150.

Tadikamalla, P.R., 1980. A look at the Burr and related distributions. Internat. Statist. Rev. 48 (3), 337–344. http://dx.doi.org/10.2307/1402945.

Vouk, M.A., 1992. Using reliability models during testing with non-operational profiles. In: Proceedings of the 2nd Bellcore/Purdue Workshop on Issues in Software Reliability Estimation. pp. 103–111.

Wood, A., 1996. Predicting software reliability. IEEE Comput. 29 (11), 69–77. http://dx.doi.org/10.1109/2.544240.

Xiao, X., Okamura, H., Dohi, T., 2012. NHPP-based software reliability models using equilibrium distribution. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 95 (5), 894–902. http://dx.doi.org/10.1587/transfun.E95.A.894.

Yamada, S., Ohba, M., Osaki, S., 1983. S-Shaped reliability growth modeling for software error detection. IEEE Trans. Reliab. R-32 (5), 475–484. http://dx.doi.org/10.1109/TR.1983.5221735.

Zhao, Y., Dohi, T., Okamura, H., 2018. Software test-run reliability modeling with non-homogeneous binomial processes. In: 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing. PRDC 2018, pp. 145–154. http://dx.doi.org/10.1109/PRDC.2018.00025.

Zhao, M., Xie, M., 1996. On maximum likelihood estimation for a general non-homogeneous Poisson process. Scand. J. Stat. 23 (4), 597–607. http://dx.doi.org/10.2307/4616427.

Zimmer, W.J., Keats, J.B., Wang, F., 1998. The Burr XII distribution in reliability analysis. J. Qual. Technol. 30 (4), 386–394. http://dx.doi.org/10.1080/00224065.1998.11979874.