A Comprehensive Evaluation for Burr-Type NHPP-based Software Reliability Models

Siqiao Li, Tadashi Dohi, Hiroyuki Okamura
Graduate School of Advanced Science and Engineering
Hiroshima University
Higashi-Hiroshima 739-8527, Japan
{rel-siqiao, dohi, okamu}@hiroshima-u.ac.jp

***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 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.**

Keywords: software reliability models, non-homogeneous Poisson processes, Burr-type distributions, goodness-of-fit performance, predictive performance.

I. 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 guarantee the quality of software. Since the software reliability is considered as one of the most fundamental and significant attributes of software quality, considerable attention has been paid to improve the software testing. At the same time, since the software testing is quite expensive, the quantification of software reliability is also another important issue in the verification phase. Since the 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 [22], [24].

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 [10], many authors proposed NHPP-based SRMs under different model assumptions. The representative NHPP-based SRMs assumed the exponential CDF [10], the gamma CDF [43], [44], the truncated-logistic CDF [26], the log-logistic CDF [12], the Pareto CDF [1], the truncated-normal CDF [28], the log-normal CDF [3], [28], the extreme-value CDFs [27] including the Weibull CDF [11].

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 to describe the software fault-detection process. Langberg and Singpurwalla [20], Miller [23], Chen and Singpurwalla [8] 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 [13] proposed a coverage based-NHPP to interpret the CDF of software fault-detection time. Huang et al. [14] found that the mean value function of NHPP-based SRMs can be represented by several kinds of algebraic mean operators. Xiao et al. [42] gave a unified modeling framework of the exponentially shaped mean value function by introducing the equilibrium distribution. Okamura and Dohi [31] 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 [1], [3], [10]–[12], [26]–[28], [43], [44] is still needed to determine the best

2767-6684/21/$31.00 ©2021 IEEE
DOI 10.1109/DSA52907.2021.00010

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 [29].

The rest of this paper is organized as follows. Section II summarizes the related work on the Burr-type SRMs. In Section III, we give 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 IV introduces the definition of the Burr-type distributions and their associated NHPP-based SRMs, where some of them are newly proposed in this paper. Section V 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. Finally, the paper is concluded with some remarks in Section VI.

II. Related Work

Burr [7] 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 [39], [45]. Abdel-Ghaly et al. [2] applied the Burr-type XII distribution to software reliability growth modeling for the first time, where they concerned the generalized order statistics SRM [23] and the Bayesian inference, and further considered an NHPP-based SRM with the Burr-type XII fault-detection time distribution for one-stage looks ahead prediction with the fault-detection time-domain data. Kim and Park [17] and Kim [18] 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 [6] 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 [22].

Prasad et al. [32] 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 [33] sequentially predicted the number of software fault counts with the same model and examined a statistical process control chart in a fashion similar to Kim [18]. Prasad et al. [34] 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 [35] 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 [16] assumed the Burr-type XII testing-effort for the NHPP-based SRM with a trend-change point.

Ahmad et al. [4], [5] 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 [36] and Chowdary et al. [9] 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 [37] compared two base-line 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 [38] to propose a different NHPP-based SRM, where the same fault-detection time-domain data as [9], [32] were analyzed. Recently, Kim [19] introduced the Burr-Hatke-exponential distribution in the NHPP-based software reliability modeling and compared it with the common exponential distribution [10] and the inverse exponential distribution with the fault-detection time-domain data.

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, it is obvious that no comprehensive comparison with the existing SRMs was made with different fault count data types. Imanaka and Dohi [15] compared the NHPP-based SRM under the Burr-type XII distribution with the representative 11 NHPP-based SRMs [1], [3], [10]–[12], [26]–[28], [43], [44] 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 reference [15] did not investigate the other Burr-type distributions and the predictive performances in the future testing period. In this paper, we develop 7 Burr-type NHPP-based SRMs, including the Burr-type III, X, and XII distributions, and compare them with the well-known NHPP-based SRMs under the time-domain data and group data circumstances. This is the first comprehensive study to evaluate the Burr-type NHPP-based SRMs and provides the empirical basis for why the Burr-type distributions are appropriate to describe the software fault-detection time distribution.

III. NHPP-based Software Reliability Modeling

A. 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;

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

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

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.

B. 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 $\boldsymbol{\theta} \in (\omega, \boldsymbol{\alpha})$. 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).

TABLE I
THE EXISTING NHPP-BASED SRMS.

Models	$\Lambda(t;\boldsymbol{\theta})$
Exponential dist. (exp) [10]	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = 1 - e^{-bt}$
Gamma dist. (gamma) [43], [44]	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \int_0^t \frac{c^b s^{b-1} e^{-cs}}{\Gamma(b)} ds$
Pareto dist. (pareto) [1]	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = 1 - (\frac{b}{t+b})^c$
Truncated normal dist. (tnorm) [28]	$\Lambda(t;\boldsymbol{\theta}) = \omega \frac{F(t;\boldsymbol{\alpha}) - F(0;\boldsymbol{\alpha})}{1 - F(0;\boldsymbol{\alpha})}$ $F(t;\boldsymbol{\alpha}) = \frac{1}{\sqrt{2\pi} b} \int_{-\infty}^t e^{-\frac{(s-c)^2}{2b^2}} ds$
Log-normal dist. (lnorm) [3], [28]	$\Lambda(t;\boldsymbol{\theta}) = \omega F(\ln t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \frac{1}{\sqrt{2\pi} b} \int_{-\infty}^t e^{-\frac{(s-c)^2}{2b^2}} ds$
Truncated logistic dist. (tlogist) [26]	$\Lambda(t;\boldsymbol{\theta}) = \omega F(t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \frac{1-e^{-bt}}{1+ce^{-bt}}$
Log-logistic dist. (llogist) [12]	$\Lambda(t;\boldsymbol{\theta}) = \omega F(\ln t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = \frac{(bt)^c}{1+(bt)^c}$
Truncated extreme-value max dist. (txvmax) [27]	$\Lambda(t;\boldsymbol{\theta}) = \omega \frac{F(t;\boldsymbol{\alpha}) - F(0;\boldsymbol{\alpha})}{1 - F(0;\boldsymbol{\alpha})}$ $F(t;\boldsymbol{\alpha}) = e^{-e^{-\frac{t-c}{b}}}$
Log-extreme-value max dist. (lxvmax) [27]	$\Lambda(t;\boldsymbol{\theta}) = \omega F(\ln t;\boldsymbol{\alpha})$ $F(t;\boldsymbol{\alpha}) = e^{-(\frac{t}{b})^{-c}}$
Truncated extreme-value min dist. (txvmin) [27]	$\Lambda(t;\boldsymbol{\theta}) = \omega \frac{F(0;\boldsymbol{\alpha}) - F(t;\boldsymbol{\alpha})}{F(0;\boldsymbol{\alpha})}$ $F(t;\boldsymbol{\alpha}) = e^{-e^{-\frac{t-c}{b}}}$
Log-extreme-value min dist. (lxvmin) [11]	$\Lambda(t;\boldsymbol{\theta}) = \omega\,(1 - F(-\ln t;\boldsymbol{\alpha}))$ $F(t;\boldsymbol{\alpha}) = e^{-e^{-\frac{t-c}{b}}}$

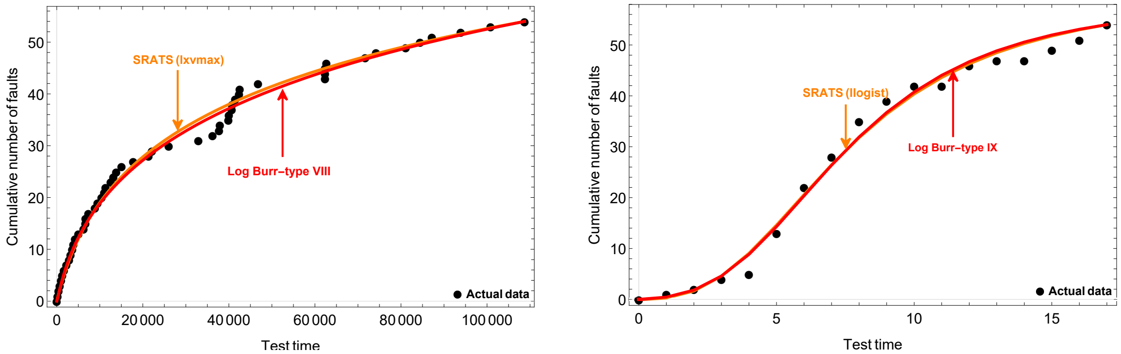
$(\omega{>}0, a{>}0, b{>}0, c{>}0)$

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 [29] 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 I, we summarize these 11 NHPP-based SRMs.

C. 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.

(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 software faults are detected, where the time sequence is given by $\boldsymbol{T} = \{t_1, t_2, \ldots, t_m\}$.



(a) TDS1.

(b) GDS1.

Fig. 1. Behavior of cumulative number of software faults with the Burr-type and existing SRMs.

Then, the likelihood function is represented as

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

so that the log likelihood function is written by

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

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, $(t_{i-1}, t_i]$ $(i = 1, 2, \ldots, m)$. 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 $\boldsymbol{I} = \{(t_i, n_i), i = 1, 2, \ldots, m\}$ are given by

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

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

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

IV. 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 [7] 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 $g(x, F(x;\boldsymbol{\alpha})) = \left(b_1 + b_2x + b_3x^2\right)^{-1}$ and $F(x;\boldsymbol{\alpha})$ 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;

$$\frac{df(x;\boldsymbol{\alpha})}{dx} = \frac{f(x;\boldsymbol{\alpha})(b_0 - x)}{(b_1 + b_2x + b_3x^2)}, \tag{11}$$

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 [7] 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 $g(x;\boldsymbol{\alpha})$ should make the CDF $F(x;\boldsymbol{\alpha})$ increase monotonously from 0 to 1 within the specified time x. The above statement is often called the *Burr hypothesis*. Finally, Burr [7] derived 12 Burr-type distributions I$\sim$XII by considering 12 kinds of $g(x;\boldsymbol{\alpha})$ functions. Table II lists the Burr-type distributions proposed in [7].

As mentioned in Section II, the Burr-type III, X, and XII distributions were applied to describe the software fault-detection time distribution in the past literature, where these CDFs have positive support $(0, \infty)$. In other words, from Table II, 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 [4], [5], [9], [36], [37], the Burr-type X distribution [38], the Burr-type XII distribution [2], [6], [16]–[18], [32]–[35] with the positive support $X \in (0, \infty)$, it is possible to

TABLE II
BURR-TYPE DISTRIBUTIONS.

Type	CDF	Domain of 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	$F(x;\boldsymbol{\alpha}) = 1-(1+x^{a})^{-b}$	$(0,+\infty)$

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

transform the CDF with support $(-\infty,+\infty)$ to the log Burr-type distributions with the support $X \in (0,\infty)$ by taking $\exp(X)$. So, we consider the log Burr-type II, VI, VII, VIII, IX distributions to represent the mean value function of the NHPP-based SRM:

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

The underlying idea comes from the log-normal NHPP-based SRM [3], [28] and the log-logistic NHPP-based SRM [12]. In fact, it is known that the logarithmic Burr-type II distribution is reduced to the log-logistic distribution [39]. Table III presents the Burr-type NHPP-based SRMs considered in this paper, where we applied a generalized Burr-type XII distribution by introducing an additional scale parameter d. That is to say, if $d = 1$, then the Burr-type XII distribution in Table III becomes the original form in Table II.

TABLE III
BURR-TYPE NHPP-BASED SRMS.

Models	CDF	$\Lambda(t;\boldsymbol{\theta})$
Burr-type III	$F(t;\boldsymbol{\alpha}) = \left(1+(t/d)^{-a}\right)^{-b}$	$\omega F(t;\boldsymbol{\alpha})$
Log Burr-type VI	$F(t;\boldsymbol{\alpha}) = \left(ae^{-c\sinh(t/d)}+1\right)^{-b}$	$\omega F(\log t;\boldsymbol{\alpha})$
Log Burr-type VII	$F(t;\boldsymbol{\alpha}) = 2^{-b}\left(1+\tanh(t/d)\right)^{b}$	$\omega F(\log t;\boldsymbol{\alpha})$
Log Burr-type VIII	$F(t;\boldsymbol{\alpha}) = (\arctan(e^{t/d})2/\pi)^{b}$	$\omega F(\log t;\boldsymbol{\alpha})$
Log Burr-type IX	$F(t;\boldsymbol{\alpha}) = 1-2\left(a\left((1+e^{t/d})^{b}-1\right)+2\right)^{-1}$	$\omega F(\log t;\boldsymbol{\alpha})$
Burr-type X	$F(t;\boldsymbol{\alpha}) = \left(1-e^{-(t/d)^2}\right)^{b}$	$\omega F(t;\boldsymbol{\alpha})$
Burr-type XII	$F(t;\boldsymbol{\alpha}) = 1-\left(\frac{1}{1+(t/d)^{a}}\right)^{b}$	$\omega F(t;\boldsymbol{\alpha})$

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

V. PERFORMANCE COMPARISONS

A. Data Sets

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

TABLE IV
TIME-DOMAIN DATA SETS.

Data	No. faults	Source
TDS1	54	SYS2 [25]
TDS2	38	SYS3 [25]
TDS3	136	SYS1 [25]
TDS4	53	—
TDS5	73	—
TDS6	38	—
TDS7	41	S27 [25]
TDS8	101	—

TABLE V
GROUP DATA SETS.

Data	No. faults	Testing days	Source
GDS1	54	17	SYS2 [25]
GDS2	38	14	SYS3 [25]
GDS3	120	19	Release2 [41]
GDS4	61	12	Release3 [41]
GDS5	9	14	NASA -supported project [40]
GDS6	66	20	DS1 [30]
GDS7	58	33	DS2 [30]
GDS8	52	30	DS3 [30]

B. Goodness-of-fit Performances

We investigate the goodness-of-fit of our 7 Burr-type NHPP-based SRMs and the existing 11 NHPP-based SRMs in SRATS [29]. 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

$$\text{AIC} = -2\ln\mathcal{L}(\hat{\boldsymbol{\theta}};\boldsymbol{T} \text{ or } \boldsymbol{I}) + 2\times(\text{the number of parameters}) \tag{14}$$

and

$$\text{MSE}(\hat{\boldsymbol{\theta}};\boldsymbol{T}) = \frac{\sqrt{\sum_{i=1}^{m}(i-\Lambda(t_i;\hat{\boldsymbol{\theta}}))^2}}{m}, \tag{15}$$

$$\text{MSE}(\hat{\boldsymbol{\theta}};\boldsymbol{I}) = \frac{\sqrt{\sum_{i=1}^{m}(n_i-\Lambda(t_i;\hat{\boldsymbol{\theta}}))^2}}{m}. \tag{16}$$

The smaller AIC/MSE is the better SRM in terms of the goodness-of-fit to the underlying fault count data.

Figure 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 7 Burr-type NHPP-based SRMs (red curve) and the existing NHPP-based SRMs in SRATS (orange 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 VI and Table VII for a more accurate comparison, respectively, where the bold font marks the best SRM with minimum AIC in each data set. From Table VI, it can be seen that in the half of time-domain data sets (TDS1, TDS2, TDS4, TDS5), 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 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 TDS2, TDS4, TDS5, and TDS8 as well.

In the group data sets, the Burr-type NHPP-based SRMs provided the smaller AIC in DS5 and DS7, and the smaller MSE in GDS7 and GDS8. 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 difference between the Burr-type NHPP-based SRM and the existing NHPP-based SRM was not observed in GDS1, GDS3, GDS4, GDS7, and GDS8. On the other hand, in GDS2 and GDS6, we notice 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. Inversely, in GDS5, the best SRATS SRM (exp) gave a better goodness-of-fit than the best Burr-type NHPP-based SRM (Log Burr-type IX) significantly. These results reinforce the conclusion by Imanaka and Dohi [15], so that the Burr-type NHPP-based SRM does not always outperform the existing NHPP-based SRMs but provides better goodness-of-fit performances in many cases.

In Tables VI and VII, 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, and TDS7, 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 five cases (GDS1, GDS2, GDS3, GDS5, and 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.

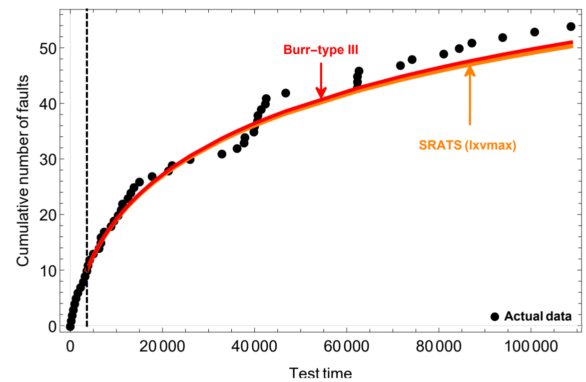
C. Predictive Performances

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 distance between the predicted cumulative number of software faults and its (unknown) realization per prediction length. Suppose that m or n_m software fault count data is available, and that the prediction length is given by $l\ (= 1, 2, \cdots)$. Then, we define the PMSE for the fault-detection time-domain data and the group data by

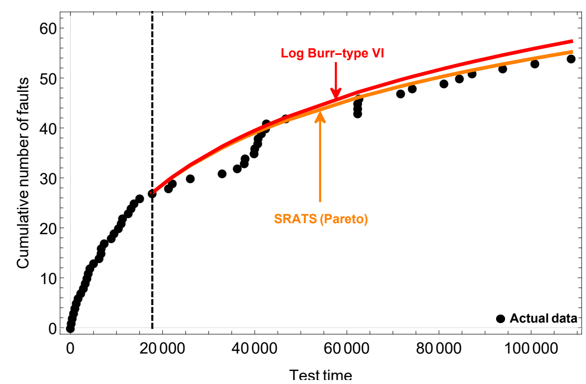
$$\mathrm{PMSE}(\hat{\boldsymbol{\theta}}; \boldsymbol{T}) = \frac{\sqrt{\sum_{i=m+1}^{m+l}\{i - \Lambda(t_i; \hat{\boldsymbol{\theta}})\}^2}}{l}, \quad (17)$$

$$\mathrm{PMSE}(\hat{\boldsymbol{\theta}}; \boldsymbol{I}) = \frac{\sqrt{\sum_{i=m+1}^{m+l}\{n_i - \Lambda(t_i; \hat{\boldsymbol{\theta}})\}^2}}{l}, \quad (18)$$

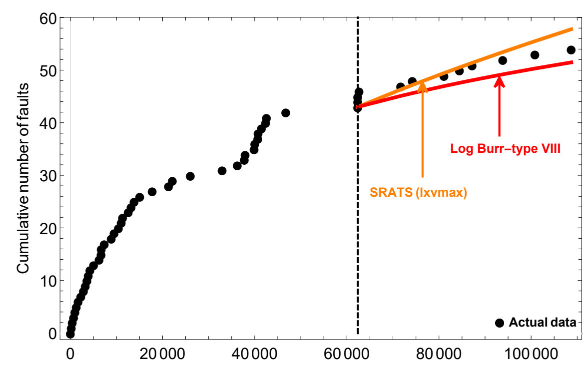
where $\hat{\boldsymbol{\theta}}$ is the ML estimated at time t_m.



(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.

TABLE VI
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)	0.190	lxvmax	896.666	232.175 (**178.175**)	0.190
TDS2	Log Burr-type VIII	**598.122**	84.883 (46.883)	**0.210**	lxvmax	598.131	82.895 (**44.895**)	0.211
TDS3	Burr-type X	1939.258	143.726 (**4.726**)	0.313	lxvmin	**1938.160**	172.526 (34.526)	**0.220**
TDS4	Burr-type III	**759.704**	54.017 (1.017)	**0.241**	pareto	759.756	53.201 (**0.201**)	0.267
TDS5	Burr-type X	**757.119**	84.808(**11.808**)	**0.481**	exp	757.869	95.960 (22.960)	0.510
TDS6	Log Burr-type VIII	722.397	66.989(28.989)	0.235	lxvmax	**721.928**	51.561 (**13.561**)	**0.195**
TDS7	Log Burr-type VII	1008.220	95.007 (**54.007**)	0.382	lxvmax	1008.220	95.796(54.796)	0.382
TDS8	Burr-type XII	2506.026	214.184(113.184)	**0.672**	pareto	**2504.170**	211.202 (**110.202**)	0.685

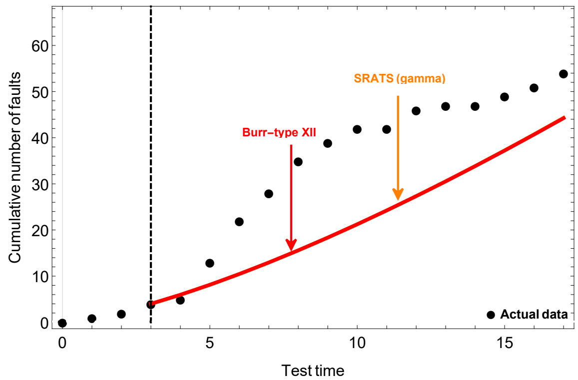
TABLE VII
Goodness-of-fit performances based on AIC (group data).

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**)	**0.474**	llogist	73.053	60.651 (6.651)	0.492
GDS2	Log Burr-type IX	**59.459**	67.468 (**29.459**)	**0.479**	lxvmax	61.694	74.334 (36.334)	0.481
GDS3	Log Burr-type VI	**85.873**	122.896 (**2.896**)	**0.403**	tnorm	87.267	123.252 (3.252)	0.569
GDS4	Log Burr-type VI	**50.600**	61.381 (0.381)	**0.165**	tlogist	51.052	62.269 (**0.269**)	0.405
GDS5	Log Burr-type IX	30.231	30.359 (**21.359**)	**0.071**	exp	**29.911**	39.052 (30.052)	0.092
GDS6	Log Burr-type IX	**103.459**	71.982(**5.983**)	**1.022**	lxvmax	108.831	134.936 (68.936)	1.061
GDS7	Burr-type III	124.767	58.178 (0.178)	0.257	txvmin	**123.265**	58.037 (**0.037**)	**0.253**
GDS8	Log Burr-type IX	**117.234**	52.516 (0.516)	0.532	llogist	117.470	52.459 (**0.459**)	0.532

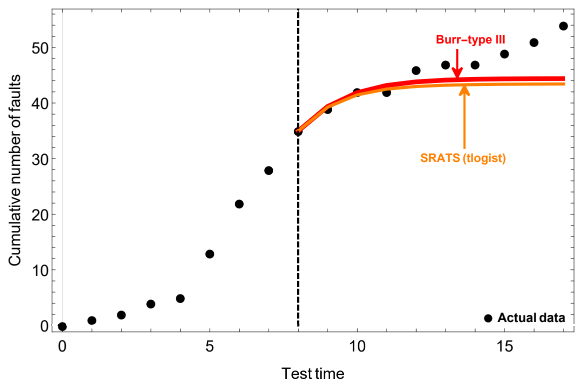
In our experiments, we set three observation points; 20%, 50%, and 80% of the whole data set, and predict the cumulative number of software faults during the remaining period, where the prediction length becomes shorter as the observation point is larger. In Figs. 2 and 3, we show the examples of predictive 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 models 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 represented the S-shaped curve, and both SRMs resulted in the miss-prediction in the early testing phases, like 20% and 50% observation points. These poor predictive performances are caused by the trend change in the future. 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 VIII and IX 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 not guarantee the smaller PMSE than SRATS SRMs in most cases of early and middle testing phases. However, when the testing phase is later (80%), the Burr-type NHPP-based SRMs provided the smaller PMSE in a total of 6 cases 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 at least half of the data sets.

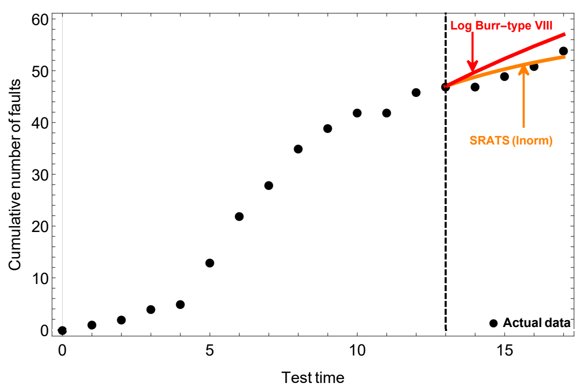
Except in the 80% observation of Table VIII, it should be noted that the best SRM with the minimum PMSE depends on the data sets in both modeling frameworks; Burr-type and SRATS. Of course, the best SRM with the minimum PMSE cannot be known in advance at each observation point. In this sense, we have to say that the comparison in Tables VIII and IX is not feasible. In Tables X and XI, 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 PMSE than the Burr-type NHPP-based SRMs in half of 8 cases (TDS1, TDS2, TDS5, and TDS6), and the Log Burr-type IX SRM outperformed the SRATS SRMs in the only TDS3. When the testing phase is middle (50%), we found that Burr-type NHPP-based SRMs did give neither the minimum AIC nor minimum PMSE. However, this worse trend can be improved in the later stage of the testing phase, so that the Burr-type NHPP-based SRMs could show the best goodness-of-fit performance in the observation



(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.

phase and ensure the minimum PMSE in the future prediction phase in TDS1, TDS3, and TDS8.

In the group data of Table XI, it can be observed that the Burr-type NHPP-based SRMs provided both the smallest AIC and smallest PMSE at the same time in some cases; *i.e.*, one case out of 8 data sets in (i), 5 cases out of 8 data sets in (ii) and (iii). These results confirm that the Burr-type NHPP-based SRMs have the higher prediction ability, especially in the late phase of software testing. 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

TABLE VIII

PREDICTIVE PERFORMANCES BASED ON PMSE (TIME-DOMAIN DATA).

(i) Prediction from the 20% observation point				
Data Set	**Burr Type**		**SRATS**	
	Best Burr	PMSE	Best SRATS	PMSE
TDS1	Burr-type III	**0.293**	lxvmax	0.332
TDS2	Burr-type XII	2.816	tnorm	**1.129**
TDS3	Log Burr-type VI	**0.506**	lxvmax	0.538
TDS4	Burr-type III	1.145	lnorm	**1.120**
TDS5	Log Burr-type VII	13.626	exp	**12.616**
TDS6	Burr-type XII	**1.253**	exp	1.595
TDS7	Log Burr-type VII	1.558	lxvmax	**0.959**
TDS8	Burr-type III	5.379	lxvmax	**4.724**
(ii) Prediction from the 50% observation point				
TDS1	Log Burr-type VI	0.577	pareto	**0.459**
TDS2	Log Burr-type IX	1.986	tlogist	**0.841**
TDS3	Log Burr-type VII	2.871	pareto	**0.409**
TDS4	Log Burr-type VII	3.620	tlogist	**1.888**
TDS5	Log Burr-type VIII	2.546	llogist	**2.229**
TDS6	Log Burr-type VIII	**0.455**	lxvmax	0.706
TDS7	Burr-type IX	22.325	exp	**13.943**
TDS8	Burr-type VI	**2.689**	lxvmax	24.711
(iii) Prediction from the 80% observation point				
TDS1	Log Burr-type VIII	**0.661**	lxvmax	0.664
TDS2	Log Burr-type VIII	**0.240**	lxvmax	0.241
TDS3	Log Burr-type VII	0.573	lxvmax	**0.560**
TDS4	Log Burr-type X	**0.468**	txvmin	0.570
TDS5	Log Burr-type VII	1.450	lxvmax	**1.128**
TDS6	Burr-type III	**0.448**	lxvmax	0.449
TDS7	Burr-type XII	**0.908**	lxvmax	0.972
TDS8	Burr-type XII	**1.416**	lxvmax	1.585

PMSEs than the SRATS SRMs in many cases; half cases in (i) and (ii), 6 cases in (iii) in Table X, 5 out of 8 data sets in (i), 7 out of 8 data sets in (ii) and 6 out of 8 sets in Table XI. We never claim here that the Burr-type NHPP-based SRMs are always better than the existing SRMs in the literature. However, we emphasize that the Burr-type NHPP-based SRMs should be the possible candidates in selecting the best model in terms of goodness-of-fit and predictive performances. Also, another new finding is that the logarithmic 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.

D. 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) $x = t_{m+l} - t_m$ or l when the software is released at time $t = t_m$. 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

$$\begin{aligned} R(x \mid t) &= \Pr(N(t+x) - N(t) = 0 \mid N(t) = m) \\ &= \exp\left(-\left[\Lambda(t+x;\boldsymbol{\theta}) - \Lambda(t;\boldsymbol{\theta})\right]\right), \end{aligned} \quad (19)$$

TABLE IX
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	**3.692**	gamma	**3.706**
GDS2	Log Burr-type VI	**0.806**	lxvmax	1.441
GDS3	Log Burr-type IX	**4.878**	gamma	6.738
GDS4	Burr-type XII	8.087	exp	**3.436**
GDS5	Log Burr-type VII	0.501	pareto	**0.432**
GDS6	Log Burr-type IX	**1.523**	tlogist	2.340
GDS7	Log Burr-type VIII	6.415	exp	**3.654**
GDS8	Burr-type XII	5.302	txvmin	**4.032**
(ii) Prediction from the 50% observation point				
GDS1	Burr-type III	**1.546**	tlogist	3.769
GDS2	Burr-type XII	2.238	txvmin	**1.835**
GDS3	Log Burr-type VIII	**5.353**	lxvmax	6.832
GDS4	Log Burr-type VIII	6.461	exp	**3.522**
GDS5	Log Burr-type IX	**0.181**	exp	0.194
GDS6	Log Burr-type IX	**5.201**	pareto	5.496
GDS7	Burr-type III	**1.088**	lxvmax	1.096
GDS8	Log Burr-type VI	1.499	txvmin	**1.306**
(iii) Prediction from the 80% observation point				
GDS1	Log Burr-type VIII	0.865	lnorm	**0.531**
GDS2	Log Burr-type VI	**0.292**	exp	0.295
GDS3	Burr-type XII	0.653	tnorm	**0.230**
GDS4	Burr-type X	**0.507**	tnorm	0.589
GDS5	Log Burr-type VI	**0.169**	tnorm	0.205
GDS6	Burr-type III	**0.523**	lnorm	0.741
GDS7	Burr-type X	1.530	txvmin	**0.818**
GDS8	Log Burr-type VIII	**0.325**	lxvmax	**0.325**

TABLE X
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	10.683	exp	**141.609**	**0.865**
TDS2	Burr-type XII	91.195	2.816	tnorm	**84.722**	**1.129**
TDS3	Log Burr-type IX	**313.458**	**2.825**	llogist	313.745	3.089
TDS4	Log Burr-type IX	126.424	**1.197**	exp	**121.858**	2.993
TDS5	Log Burr-type IX	114.575	25.328	exp	**113.372**	**12.616**
TDS6	Burr-type X	129.216	2.757	lxvmax	**128.656**	**2.250**
TDS7	Log Burr-type VIII	189.539	**1.605**	exp	**187.583**	4.763
TDS8	Log Burr-type IX	441.559	**18.276**	exp	440.510	41.053
(ii) Prediction from the 50% observation point						
TDS1	Burr-type X	403.494	2.400	exp	**403.368**	**1.890**
TDS2	Log Burr-type VIII	256.977	3.231	exp	**256.074**	**0.841**
TDS3	Log Burr-type IX	**861.677**	3.951	llogist	861.949	**3.704**
TDS4	Log Burr-type IX	**334.737**	4.770	exp	334.762	**1.913**
TDS5	Log Burr-type IX	364.639	**2.822**	exp	**363.831**	**3.128**
TDS6	Log Burr-type VIII	344.810	**0.455**	lxvmax	**344.604**	0.706
TDS7	Burr-type X	447.125	**33.358**	tlogist	**445.247**	412.610
TDS8	Burr-type X	1094.480	**55.801**	exp	**1092.710**	66.558
(iii) Prediction from the 80% observation point						
TDS1	Log Burr-type VIII	**691.675**	**0.661**	lxvmax	691.677	0.664
TDS2	Log Burr-type VIII	443.895	**0.240**	lxvmax	**443.891**	0.241
TDS3	Log Burr-type IX	**1478.400**	**0.921**	llogist	1478.500	0.943
TDS4	Log Burr-type X	566.040	**0.468**	exp	565.497	0.570
TDS5	Log Burr-type IX	**577.096**	1.907	llogist	**577.358**	**1.740**
TDS6	Log Burr-type VII	553.819	0.751	lxvmax	**553.472**	**0.449**
TDS7	Log Burr-type IX	770.108	**1.272**	exp	**769.836**	1.366
TDS8	Burr-type III	**1887.900**	**2.534**	pareto	**1889.040**	2.536

TABLE XI
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	Log Burr-type VIII	12.737	**4.383**	exp	**11.085**	6.241
GDS2	Log Burr-type IX	**12.865**	**1.190**	lxvmax	**12.865**	1.441
GDS3	Log Burr-type IX	18.976	**4.878**	exp	**18.442**	10.149
GDS4	Log Burr-type VII	12.649	8.197	exp	**11.669**	**3.436**
GDS5	Log Burr-type VII	8.000	0.501	exp	**7.386**	**0.432**
GDS6	Log Burr-type VII	20.744	**4.715**	lnorm	**20.660**	5.147
GDS7	Log Burr-type III	17.340	**6.566**	txvmin	**16.958**	6.600
GDS8	Burr-type X	**8.614**	8.313	txvmin	**8.614**	**4.032**
(ii) Prediction from the 50% observation point						
GDS1	Log Burr-type IX	**36.015**	**3.902**	lxvmin	36.846	4.346
GDS2	Log Burr-type IX	**29.849**	**2.311**	lxvmax	31.051	2.633
GDS3	Log Burr-type VI	**48.466**	**6.776**	exp	49.313	9.316
GDS4	Log Burr-type IX	31.353	**11.040**	tlogist	**30.560**	22.973
GDS5	Log Burr-type IX	17.787	**0.181**	exp	**17.365**	0.194
GDS6	Log Burr-type IX	**38.584**	**5.201**	lxvmax	40.521	5.634
GDS7	Log Burr-type IX	**71.671**	**1.094**	lxvmax	72.390	1.096
GDS8	Log Burr-type VI	65.543	1.499	txvmin	**65.835**	**1.306**
(iii) Prediction from the 80% observation point						
GDS1	Log Burr-type VIII	**55.320**	**0.865**	lxvmin	56.861	1.390
GDS2	Burr-type IX	**50.295**	**0.396**	lxvmax	52.523	0.417
GDS3	Log Burr-type VI	**73.981**	**0.708**	txvmin	75.292	0.887
GDS4	Log Burr-type VI	43.129	**0.507**	txvmin	**42.540**	0.828
GDS5	Log Burr-type IX	24.517	0.612	exp	**24.271**	**0.286**
GDS6	Burr-type XII	**93.323**	**0.523**	lxvmax	96.179	0.944
GDS7	Log Burr-type IX	**112.543**	3.603	txvmin	112.836	**0.818**
GDS8	Log Burr-type III	**100.325**	**0.834**	tlogist	**100.325**	0.855

where m is the cumulative number of software faults detected up to time t in the time-domain data (m in Eq.(19) is replaced by n_m in the group data). In our example, we suppose that the prediction length x is equivalent to the testing length experienced before, say, $t = x$.

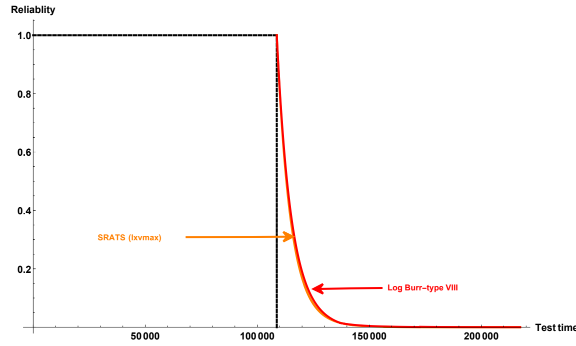
Tables XII and XIII present the quantitative software reliability, where we assume the Burr-type NHPP-based SRM and the SRATS SRM with the minimum AIC, in the fault-detection time-domain and group data sets, respectively, where the bold font denotes the case with greater reliability estimate. Looking 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 the half of time-domain data sets and 5 out of 8 group data sets. This feature tells us that the Burr-type NHPP-based SRMs tend to make more optimistic decisions in software reliability assessment than the SRATS 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. Figure 4 (a) and (b) show the software reliability estimates with the Burr-type NHPP-based SRM and the SRATS 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 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.

TABLE XII
SOFTWARE RELIABILITY ASSESSMENT WITH THE BEST AIC (TIME-DOMAIN DATA).

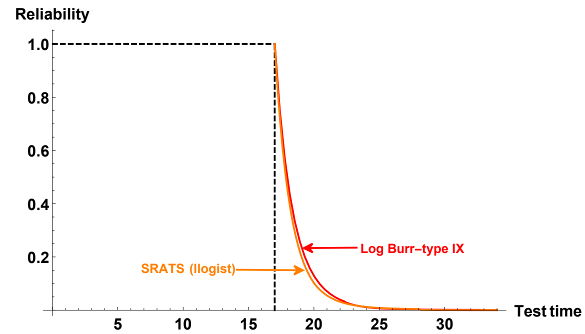
	Burr Type		SRATS	
	Best Burr	**Reliability**	**Best SRATS**	**Reliability**
TDS1	Log Burr-type VIII	2.631E-06	lxvmax	**2.674E-06**
TDS2	Log Burr-type VIII	3.687E-03	lxvmax	**3.751E-03**
TDS3	Burr-type X	**4.592E-05**	lxvmin	2.516E-10
TDS4	Burr-type III	4.573E-01	pareto	**1.000E-00**
TDS5	Burr-type X	**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	Burr-type XII	**8.971E-06**	pareto	7.736E-06

TABLE XIII
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	Log Burr-type VI	**7.119E-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



(a) TDS1.



(b) GDS1.

Fig. 4. Predictive software reliability assessment with the best Burr-type and SRATS NHPP-based SRMs.

VI. 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. More specifically, the Burr type NHPP-based SRMs have provided lower AICs in most data sets (10 out of 16 cases on AIC and MSE) and lower PMSEs in the half of group data sets used in the analysis. Our results suggest that the Burr-type NHPP-based SRMs are quite attractive SRMs to describe the software fault-detection processes and have a higher potential in 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 [29] 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 7 Burr-type NHPP-based SRMs. In addition to the logarithmic Burr-type NHPP-based SRMs, it is pointed out that the truncation at the origin enables us to apply the Burr VI, VII, VIII, IX distributions with the support $(-\infty, +\infty)$. In the subsequent paper, we will investigate the goodness-of-fit and predictive performances for such truncated Burr-type NHPP-based SRMs.

ACKNOWLEDGEMNT

This paper is the full paper version of an extended abstract [21] presented by the first author (S.L.). Also the second author is supported by JSPS KAKENHI Grant Number 19K04905.

REFERENCES

[1] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood, "Evaluation of competing software reliability predictions," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 9, pp. 950–967, 1986.
[2] A. A. Abdel-Ghaly, G.R. Al-Dayian, F.H. Al-Kashkari, "The use of Burr-type XII distribution on software reliability growth modelling," *Microelectronics and Reliability*, vol. 37, no. 2, pp. 305–313, 1997.
[3] J. A. Achcar, D. K. Dey, and M. Niverthi, "A Bayesian approach using nonhomogeneous Poisson processes for software reliability models," in *Frontiers in Reliability*, A. P. Basu, K. S. Basu, and S. Mukhopadhyay (eds.), pp. 1–18, World Scientific, Singapore, 1998.
[4] N. Ahmad, M. G. M. Khan, S. M. K. Quadri and M. Kumar, "Modelling and analysis of software reliability with Burr-type III testing-effort and release time determination," *Journal of Modelling in Management*, vol. 4, no. 1, pp. 28–54, 2009.
[5] N. Ahmad, S. M. K. Quadri, M. G. N. Khan and M. Kumar, "Software reliability growth models incorporating Burr-type III test-effort and cost-reliability analysis," *International Journal of Computer Science and Information Technologies*, vol. 2, no. pp.555–562, 2011.
[6] J. H. An, "Two model comparisons of software reliability analysis for Burr-type XII distribution," *Journal of the Korean Data & Information Science Society*, vol. 23, no. 4, pp. 815–823, 2012.

[7] I. W. Burr, “Cumulative frequency functions,” *Annals of Mathematical Statistics*, vol. 13, no. 2, pp. 215–232, 1942.

[8] Y. Chen, and N. D. Singpurwalla, “Unification of software reliability models by self-exciting point processes,” *Advances in Applied Probability*, vol. 29, pp. 337–352, 1997.

[9] C. S. Chowdary, R. S. Prasad, and K. Sobhana, “Burr-type III software reliability growth model,” *IOSR Journal of Computer Engineering*, vol. 17, no. 1, pp. 49–54, 2015.

[10] A. L. Goel, and K. Okumoto, “Time-dependent error-detection rate model for software reliability and other performance measures,” *IEEE Transactions on Reliability*, vol. R-28, no. 3, pp. 206–211, 1979.

[11] A. L. Goel, “Software reliability models: assumptions, limitations and applicability,” *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, pp. 1411–1423, 1985.

[12] S. S. Gokhale, and K. S. Trivedi, “Log-logistic software reliability growth model,” *Proceedings of the 3rd IEEE International Symposium on High-Assurance Systems Engineering (HASE-1998)*, pp. 34–41, IEEE CPS, 1998.

[13] S. S. Gokhale, and K. S. Trivedi, “A time/structure based software reliability model,” *Annals of Software Engineering*, vol. 8, pp. 85–121, 1999.

[14] C. Y. Huang, M. R. Lyu and S. Y. Kuo, “A unified scheme of some nonhomogeneous Poisson process models for software reliability estimation,” *IEEE Transactions on Software Engineering*, vol. 29, pp. 261–269, 2003.

[15] T. Imanaka, and T. Dohi, “Software reliability modeling based on Burr XII distributions,” *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98-A, no. 10, pp. 2091–2095, 2015.

[16] S. F. Islam, “Introducing Burr-type XII testing-effort with change point based software reliability growth model,” *Global Scientific Journals*, vol. 8, no. 8, pp. 2712–2718, 2020.

[17] H.-C. Kim, and H.-K. Park, “The comparative study of software optimal release time based on Burr distribution,” *International Journal of Advancements in Computing Technology*, vol. 2, no. 3, pp. 119–128, 2010.

[18] H.-C. Kim, “Assessing software reliability based on NHPP using SPC,” *International Journal of Software Engineering and Its Applications*, vol. 7, no. 6, pp. 61–70, 2013.

[19] H.-C. Kim, “A study on comparative evaluation of software reliability model applying modified exponential distribution,” *International Journal of Engineering Research and Technology*, vol. 15, no. 5, pp. 867–872, 2020.

[20] N. Langberg, and N. D. Singpurwalla, “Unification of some software reliability models,” *SIAM Journal of Scientific Computing*, vol. 6, pp. 781–790, 1985.

[21] S. Li, “A useful parametric family to characterize NHPP-based software reliability models,” *Proceedings of the 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2021)*, Doctoral Forum, IEEE CPS, 2021.

[22] M. R. Lyu (ed.), *Handbook of Software Reliability Engineering*, McGraw-Hill, New York, 1996.

[23] D. R. Miller, “Exponential order statistic models of software reliability growth,” *IEEE Transactions on Software Engineering*, vol. SE-12, pp. 12–24, 1986.

[24] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability, Measurement, Prediction, Application*, McGraw-Hill, New York, 1987.

[25] J. D. Musa, “Software reliability data,” *Technical Report in Rome Air Development Center*, 1979.

[26] M. Ohba, “Inflection S-shaped software reliability growth model,” *Stochastic Models in Reliability Theory*, S. Osaki and Y. Hatoyama (eds.), pp. 144–165, Springer-Verlag, Berlin, 1984.

[27] K. Ohishi, H. Okamura, and T. Dohi, “Gompertz software reliability model: estimation algorithm and empirical validation,” *Journal of Systems and Software*, vol. 82, no. 3, pp. 535–543, 2009.

[28] H. Okamura, T. Dohi, and S. Osaki, “Software reliability growth models with normal failure time distributions,” *Reliability Engineering and System Safety*, vol. 116, pp. 135–141, 2013.

[29] H. Okamura, and T. Dohi, “SRATS: software reliability assessment tool on spreadsheet,” *Proceedings of the 24th International Symposium on Software Reliability Engineering (ISSRE-2013)*, pp. 100–117, IEEE CPS, 2013.

[30] H. Okamura, Y. Etani, and T. Dohi, “Quantifying the effectiveness of testing efforts on software fault detection with a logit software reliability growth model,” *Proceedings of 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, IEEE CPS, 2011.

[31] H. Okamura, and T. Dohi, “Phase-type software reliability model: parameter estimation algorithms with grouped data,” *Annals of Operations Research*, vol. 244, no. 1, pp. 177–208, 2016.

[32] R. S. Prasad, K. V. M. Mohan and G. Sridevi, “Burr-type XII software reliability growth model,” *International Journal of Computer Applications*, vol. 108, no. 16, pp. 16–20, 2014.

[33] R. S. Prasad, K. V. M. Mohan and G. Sridevi, “Monitoring Burr-type XII software quality using SPC,” *International Journal of Applied Engineering Research*, vol. 9, no. 22, pp. 16651–16660, 2014.

[34] R. S. Prasad, B. R. Devi and G. Sridevi, “Assessing Burr-type XII software reliability for interval domain data using SPC,” *Computer Engineering*, vol. 79, pp. 30335–30340, 2015.

[35] M. S. Ravikumar, and R. R. L. Kantam, “Software reliability model based on Burr-type XII distribution,” *International Journal of Advanced Engineering Research and Applications*, vol. 2, no. 9, pp. 561–564, 2017.

[36] K. Sobhana, and R. S. Prasad, “Burr-type III software reliability with SPC - An order statistics approach,” *International Journal of Research Studies in Computer Science and Engineering*, vol. 2, no. 3, pp. 21–24, 2015.

[37] G. Sridevi, and C. M. S. Rani, “Comparison of software reliability analysiss for Burr distribution,” *Journal of Theoretical and Applied Information Technology*, vol. 81, no. 1, pp. 144–150, 2015.

[38] G. Sridevi, and S. Akbar, “Burr-type X software reliability growth model,” *Asian Journal of Information Technology*, vol. 15, no. 16, pp.2988–2991, 2016.

[39] P. R. Tadikamalla, “A look at the Burr and related distributions,” *International Statistical Review*, vol. 48, no. 3, pp. 337–344, 1980.

[40] M. A. Vouk, “Using reliability models during testing with non-operational profile,” *Proceedings of the 2nd Bell-core/Purdue Workshop on Issues in SoftwareReliability Estimation*, pp. 254–266, 1992.

[41] A. Wood, “Predicting software reliability,” *IEEE Computer*, vol. 20, no. 11, pp. 69–77, 1996.

[42] X. Xiao, H. Okamura and T. Dohi, “NHPP-based software reliability models using equilibrium distribution,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (A)*, vol. E95-A, no. 5, pp. 894–902, 2012.

[43] S. Yamada, M. Ohba, and S. Osaki, “S-shaped reliability growth modeling for software error detection,” *IEEE Transactions on Reliability*, vol. R-32, no. 5, pp. 475–478, 1983.

[44] M. Zhao, and M. Xie, “On maximum likelihood estimation for a general non-homogeneous Poisson process,” *Scandinavian Journal of Statistics*, vol. 23, no. 4, pp. 597–607, 1996.

[45] W. J. Zimmer, J. B. Keats and F. K. Wang, “The Burr XII distribution in reliability analysis,” *Journal of Quality Technology*, vol. 30. no. 4, pp. 386–394, 2018.