On the Useful Class of Hawkes Processes with Application to Software Reliability Engineering *

Nanxiang Qiu†, Tadashi Dohi, Hiroyuki Okamura
Graduate School of Advanced Sciences and Engineering
Hiroshima University

1 Introduction

Software reliability is an essential measure for evaluating the quality and robustness of software systems. By quantifying the likelihood that software will perform without failure during a designated operational period, software reliability models (SRMs) offer critical insights that support effective quality assurance. Over the years, a myriad of statistical and stochastic approaches have been developed to model fault occurrence during the testing, each aiming to better capture the uncertainties inherent in software systems [1, 2, 3, 4].

SRMs based on the Non-homogeneous Poisson Process (NHPP) [5, 6] have been particularly influential, as they allow failure rates to evolve over time. This adaptability makes NHPP-based SRMs to describe stochastic processes whose intensity functions are not constant—a characteristic often observed during the software testing lifecycle.

In an effort to capture additional complexities, the Hawkes process (HKP) has been employed to model self-exciting phenomena [7, 8, 9]. In its conventional form, the Hawkes process assumes a constant baseline intensity and incorporates excitation function called kernel function, to encapsulate how historical events stimulate future occurrences. This self-reinforcing behavior aligns well with scenarios in software reliability, where the occurrence of a software fault can trigger subsequent faults or expose latent faults. Recently, we tried to use a mathematical method to obtain the mean value function of the HKP-based SRMs.

In this article, we introduce a new class of HKP-based SRMs constructed using four distinct kernel functions. Initially, experiments with eight different fault-detection time datasets using the simple HKP-based SRMs showed that their predictive performances were inferior to those of NHPP-based SRMs. Motivated by these findings, we improve the model by allowing the baseline intensity to contain both time-dependent part and a specific probability density function. This modeling framework leads to a marked enhancement in prediction accuracy, showcasing the potential benefits of incorporating more complex intensity dynamics in software reliability modeling.

2 NHPP-based Software Reliability Modeling

The Non-Homogeneous Poisson Process (NHPP) is a versatile stochastic process widely used to model discrete events over time, particularly when the event rate varies. In software reliability, NHPP provides a natural framework for describing fault detection during the testing, where the detection rate typically decreases as faults are uncovered and fixed.

2.1 Statistical Properties

Let $N(t)$ represent the cumulative number of software faults detected by time t. The probability mass function (PMF) of $N(t)$ is given by

$$P(N(t) = n) = \frac{[\Lambda(t;\boldsymbol{\theta})]^n}{n!} e^{-\Lambda(t;\boldsymbol{\theta})}, \quad n = 0, 1, 2, \dots, \tag{1}$$

where $\Lambda(t;\boldsymbol{\theta}) = \int_0^t \lambda(s;\boldsymbol{\theta})\, ds$ is the mean value function, representing the expected cumulative number of software faults detected by time t.

2.2 NHPP-based SRMs

The NHPP-based software reliability models (SRMs) assume that software faults are detected randomly over time, following a cumulative distribution function (CDF) $F(t;\boldsymbol{\pi})$, where $\boldsymbol{\pi}$ denotes the distribution parameters, where the fault-detection are independent and identically distributed.

The mean value function for the NHPP-based SRMs is expressed as:

$$\Lambda(t; \boldsymbol{\theta}) = \omega F(t; \boldsymbol{\pi}),$$

where ω is the expected initial number of software faults, and $F(t; \boldsymbol{\pi})$ describes the fault detection time distribution function. As $t \to \infty$, $\Lambda(t; \boldsymbol{\theta}) \to \omega$, reflecting the finite number of detectable faults as $F(t; \boldsymbol{\pi}) \to 1$.

In this article, we assume four common NHPP-based SRMs, each defined by a specific CDF $F(t; \boldsymbol{\pi})$ in Table 1.

Table 1: Representative mean value functions of NHPP-based SRMs.

Models	**CDF** ($F(t; \boldsymbol{\pi})$)	**Mean Value Function** ($\Lambda(t; \boldsymbol{\theta})$)
Exponential (exp)	$F(t) = 1 - e^{-bt}$	$\Lambda(t) = \omega F(t)$
Gamma (gamma)	$F(t) = \frac{\int_0^t c^b s^{b-1} e^{-cs} ds}{\Gamma(b)}$	$\Lambda(t) = \omega F(t)$
Pareto (pareto)	$F(t) = 1 - \left(\frac{b}{t+b}\right)^c$	$\Lambda(t) = \omega F(t)$
Weibull (weibull)	$F(t) = e^{-e^{-(t-c)/b}}$	$\Lambda(t) = \omega(1 - F(t))$

3 Hawkes Processes

3.1 Mathematical Definition of Hawkes Processes

The key characteristic of a Hawkes process is its conditional intensity function $\lambda(t)$, which represents the instantaneous rate of event occurrence at time t, given the history of events up to that point. It is mathematically expressed as

$$\lambda(t) = \mu + \int_0^t \phi(t - s) dN(s), \tag{2}$$

where

- μ is the baseline constant intensity, representing the rate of events that occur independently of past history,
- $\phi(t - s)$ is the excitation function (or kernel function), describing how an event at time s contributes to the intensity at time t,
- $dN(s)$ denotes increments of the counting process $N(t)$, which tracks the cumulative number of events.

Equivalently, the intensity function can be expressed in summation form, highlighting the discrete contributions of past events as

$$\lambda(t) = \mu + \sum_{T_i < t} \phi(t - T_i), \tag{3}$$

where T_i denotes the occurrence times of past events before t. This formulation emphasizes the self-exciting nature of the process, where each event adds to the intensity through the kernel function.

3.2 Excitation Functions

The excitation function $\phi(t)$ quantifies the influence of past events on the current intensity. It typically adopts a parametric form scaled by a factor $\alpha(> 0)$, which controls the magnitude of the influence. In this article, we examine four distinct kernel functions that correspond to the four different NHPP-based SRMs. These kernel functions, which play a crucial role in determining the self-exciting dynamics of the Hawkes processes, are presented in Table 2.

Table 2: Representative kernels of HKP-based SRMs.

Kernels	$\phi(t;\boldsymbol{\pi})$, $\boldsymbol{\pi} = (b, c)$
Exponential (exp)	αe^{-bt}
Gamma (gamma)	$\alpha \frac{b^c t^{c-1} e^{-bt}}{\Gamma(c)}$
Pareto (pareto)	$\alpha \frac{bc^b}{(c+t)^{b+1}}$
Weibull (weibull)	$\alpha \frac{b}{c^b} t^{b-1} e^{-(t/c)^b}$

3.3 Mean Value Functions

The mean value function $E[N(t)]$ plays a crucial role in capturing the temporal dynamics of the Hawkes process. Deriving its closed-form expression is challenging due to the recursive structure of the process. To tackle this, we adopt the method proposed by Cui et al. [10], which leverages a renewal equation framework to derive the mean value function for specific kernel types. Depending on the kernel's properties, the method yields either explicit expressions or integral-differential equations.

3.3.1 Derivation Procedure

The approach defines a general form $f(t) = E[g(N(t), \lambda(t), t)]$, where an arbitrary function $g(\cdot)$ involves the counting process $N(t)$ and intensity $\lambda(t)$. By considering small increments Δt, the method derives differential equations by

1. Approximating event probabilities in $[t, t + \Delta t]$ using the process history $\mathcal{F}_t$,
2. Expanding $g(\cdot)$ via a Taylor series and taking the limit $\Delta t \to 0$,
3. Solving the resulting equations analytically or numerically.

3.3.2 Results

Using the above procedure, we can derive the mean value functions $E[N(t)]$ for exponential and gamma kernel functions. For the exponential kernel $\phi(t) = \alpha e^{-bt}$, the mean value function takes the form:

$$E_{\text{Exp}}[N(t)] = \frac{-b\mu}{\alpha - b} + \frac{\alpha\mu}{(\alpha - b)^2}\left[e^{(\alpha-b)t} - 1\right], \tag{4}$$

where μ represents the constant baseline intensity and $\alpha, b(> 0)$ are kernel parameters.

For the gamma kernel (with $c = 2$), we have

$$E_{\text{Gamma}}[N(t)] = \frac{\mu t}{1-\alpha} + \frac{\mu\sqrt{\alpha b}}{2(1+\sqrt{\alpha})^2}\left[1 - e^{-\frac{1+\sqrt{\alpha}}{b}t}\right] - \frac{\mu\sqrt{\alpha b}}{2(1-\sqrt{\alpha})^2}\left[1 - e^{-\frac{1-\sqrt{\alpha}}{b}t}\right]. \tag{5}$$

However, the HKP-based SRMs with Pareto and Weibull kernels, are more complex to derive explicit expressions of $E[N(t)]$. Hence, it is common to solve differential equations numerically. For the Pareto kernel and the Weibull kernel, the differential-integral equations are given by

$$\text{Pareto: } \frac{d}{dt}E[N(t)] = \mu + \int_0^t \alpha \frac{bc^b}{(c+t-s)^{b+1}} E[N(s)]\, ds, \tag{6}$$

$$\text{Weibull: } \frac{d}{dt}E[N(t)] = \mu + \int_0^t \alpha \frac{b}{c^b}(t-s)^{b-1} e^{-((t-s)/c)^b} E[N(s)]\, ds, \tag{7}$$

respectively.

4 Experiments

4.1 Parameter Estimation

To estimate the model parameters in both NHPP-based SRMs and HKP-based SRMs, we employ the maximum likelihood estimation method. This approach is fundamental for ensuring that the model parameters align with the observed software fault count data. For the NHPP and HKP -based SRMs, the parameter vectors are defined as $\theta_{\text{NHPP}} = (\omega, b, c)$ and $\theta_{\text{HKP}} = (\mu, \alpha, b, c)$, respectively.

Assuming $t_0 = 0$, the detection time for each software fault is recorded, resulting in n distinct time points. The general form of the log-likelihood function for these processes, given event observations over the time interval $[0, T]$, is expressed as

$$\ln L(\theta) = \sum_{i=1}^{n} \log \lambda(t_i; \theta) - \int_0^T \Lambda(s; \theta)\, ds, \tag{8}$$

where $\lambda(t; \theta)$ represents the intensity function that depends on the parameter vector θ, and the cumulative intensity function is defined as

$$\Lambda(T; \theta) = \int_0^T \lambda(s; \theta)\, ds. \tag{9}$$

4.2 Data Sets

Table 3 summarizes the software fault-detection time datasets used in our experiments. These datasets vary in the number of faults and maximum testing duration, providing diverse scenarios for evaluating the models.

4.3 Predictive Performance

4.3.1 Akaike Information Criterion (AIC)

The AIC is a widely used measure for model selection, balancing goodness-of-fit and model complexity. The AIC is defined as:

$$\text{AIC} = -2 \ln L(\hat{\boldsymbol{\theta}}) + 2 \times (\text{number of parameters}), \tag{10}$$

where $\ln L(\hat{\boldsymbol{\theta}})$ is the log-likelihood function maximized on the parameter space. A smaller AIC value is preferable, indicating a model with a better goodness-of-fit per-

Table 3: Software fault-detection time data sets.

Data	No. faults	Maximum testing time	Source
TDS1	54	108708	SYS2[2]
TDS2	24	1095.88	SYS4[2]
TDS3	41	4312598	SYS27[2]
TDS4	129	89040	CSR2[1]
TDS5	197	50236822	SYS4[2]
TDS6	136	88682	SYS1[2]
TDS7	104	15369.5	SRC3[1]
TDS8	397	108890	SRC1[1]

formance.

4.3.2 Predictive Mean Squared Error (PMSE)

The PMSE evaluates the model's predictive accuracy over future fault counts. It is defined as

$$\mathrm{PMSE} = \sqrt{\frac{\sum_{i=p+1}^{m} \left(E[N(t_i)] - n_i\right)^2}{m-p}}, \tag{11}$$

where

- $p \sim (= 0, 1, \ldots, m-1)$: the number of observation points,
- m: the total number of fault count data points in the future,
- $E[N(t_i)]$: the expected cumulative number of faults at time t_i.
- n_i: the observed fault counts at t_i.

Also, a smaller PMSE indicates better predictive performance for SRMs. Note that the model selection is based on the AIC, as the minimum PMSE cannot be known at each observation point in advance. The model with the lowest AIC at each observation point is chosen for prediction, and its PMSE is evaluated during the remaining prediction period.

To evaluate the models' predictive capabilities, each dataset is partitioned into segments representing different stages of software testing. Specifically, the datasets are split into three distinct proportions; 20%, 50%, and 80%. These segments are used as training data, with the remaining portion reserved for prediction testing.

Table 4: Predictive performances based on SRMs with minimum AIC.

Dataset	Length	Hawkes SRMs				NHPP SRMs			
		Exp	Gamma	Pareto	Weibull	Exp	Gamma	Pareto	Weibull
TDS1	20%	131.811 (88.781)	143.683 (15.254)	145.750 (100.301)	145.683 (100.327)	141.609 (0.885)	143.607 (1.371)	143.597 (0.984)	143.609 (1.047)
	50%	397.380 (46.655)	410.396 (10.146)	412.396 (48.111)	412.396 (48.135)	403.368 (1.960)	405.321 (1.763)	405.241 (0.422)	405.307 (1.657)
	80%	703.454 (10.410)	718.037 (3.050)	720.037 (10.003)	713.458 (8.686)	693.172 (1.562)	693.175 (1.248)	692.119 (0.792)	692.911 (1.147)
TDS2	20%	32.155 (3.428)	37.702 (0.614)	45.379 (6.727)	40.827 (41792.242)	41.394 (0.779)	36.459 (2.491)	43.394 (0.781)	37.079 (2.393)
	50%	108.921 (3.704)	110.400 (0.625)	127.456 (4.076)	121.137 (5.904)	123.477 (1.200)	125.241 (5.432)	125.476 (1.197)	125.242 (1.077)
	80%	148.356 (1.483)	134.219 (2.254)	184.396 (2.904)	148.747 (3.317)	180.401 (1.487)	178.957 (3.694)	182.401 (1.491)	178.958 (3.680)
TDS3	20%	168.505 (23.178)	189.583 (4.930)	191.583 (28.013)	191.583 (28.067)	185.583 (4.902)	189.587 (5.451)	189.583 (4.931)	189.584 (4.844)
	50%	428.761 (63.906)	448.960 (14.623)	450.960 (66.998)	450.960 (67.017)	446.961 (14.602)	447.187 (29.795)	448.960 (14.620)	447.126 (34.698)
	80%	765.838 (5.450)	788.317 (1.911)	790.317 (5.259)	790.317 (5.245)	769.836 (1.516)	771.833 (1.508)	770.823 (1.247)	771.754 (1.462)
TDS4	20%	296.611 (160.580)	302.042 (25.874)	308.678 (168.496)	308.678 (168.891)	304.679 (16.509)	292.293 (5.418)	306.678 (16.548)	293.297 (5.642)
	50%	759.235 (213.845)	766.993 (41.926)	777.708 (212.041)	746.933 (98.901)	773.710 (26.408)	766.999 (2.339)	775.708 (26.479)	767.914 (3.012)
	80%	1415.779 (57.985)	1410.700 (22.886)	1430.865 (65.843)	1338.791 (15.437)	1392.340 (1.196)	1394.180 (1.395)	1389.400 (0.713)	1394.250 (1.036)
TDS5	20%	1024.168 (6.354)	1048.920 (0.412)	1050.923 (6.861)	1045.598 (10.841)	1046.920 (0.414)	1048.910 (0.423)	1048.920 (0.412)	1048.920 (0.396)
	50%	2580.375 (16.665)	2604.900 (1.773)	2606.905 (16.540)	2592.648 (5.317)	2602.910 (1.764)	2603.980 (2.859)	2604.900 (1.771)	2603.990 (2.808)
	80%	4180.704 (4.937)	4205.460 (0.771)	4207.462 (5.046)	4147.627 (4.672)	4203.050 (0.700)	4204.780 (0.752)	4205.060 (0.699)	4204.770 (0.762)
TDS6	20%	309.677 (214.801)	318.555 (24.650)	321.319 (225.236)	321.319 (224.576)	313.856 (5.214)	313.746 (3.179)	314.873 (2.826)	313.746 (2.366)
	50%	866.581 (108.194)	877.733 (14.250)	880.385 (112.026)	871.833 (93.968)	866.494 (2.610)	861.997 (2.966)	866.128 (0.430)	861.954 (3.600)
	80%	1528.027 (24.887)	1539.150 (5.139)	1542.242 (24.564)	1520.136 (13.839)	1487.630 (1.229)	1478.740 (1.347)	1481.450 (1.125)	1478.560 (1.150)
TDS7	20%	184.206 (144.529)	188.847 (20.621)	195.229 (134.113)	195.229 (134.151)	191.227 (10.691)	193.172 (11.623)	191.227 (10.573)	193.170 (11.468)
	50%	538.462 (75.412)	537.187 (16.582)	543.787 (78.307)	527.477 (42.827)	527.080 (3.051)	528.617 (3.399)	529.080 (3.049)	528.395 (3.598)
	80%	926.083 (11.670)	925.070 (0.805)	922.125 (0.560)	924.394 (0.717)	926.110 (1.150)	923.704 (0.896)	920.113 (0.600)	924.334 (1.110)
TDS8	20%	758.928 (635.488)	757.683 (52.369)	768.582 (642.936)	764.886 (395.179)	764.584 (35.906)	756.954 (7.989)	766.582 (36.059)	755.616 (9.443)
	50%	1899.158 (820.466)	1868.690 (94.544)	1908.763 (824.218)	1881.202 (554.593)	1904.770 (58.493)	1901.640 (13.894)	1906.760 (58.678)	1901.820 (14.161)
	80%	3695.260 (230.882)	3617.410 (62.429)	3706.932 (230.906)	3420.119 (5.934)	3489.310 (4.017)	3489.140 (4.309)	3485.260 (2.613)	3491.100 (4.137)

Table 4 presents the predictive performances when the baseline model was selected with the minimum AIC at each observation point with time-domain data, AIC values are shown outside the parentheses, while PMSE values are inside. From the results in Table 4, in certain instances, the Hawkes process-based SRMs achieved lower AIC values, suggesting their potential for effectively modeling software fault-detection time data. Nonetheless, the predictive performance of the Hawkes process-based SRMs was generally subpar in the majority of cases.

5 Alternative Hawkes process-based SRMs

We find that in the vast majority of cases, the prediction function curve for this form of simple autoregressive model based on the Hawkes process is usually a linearly increasing straight line, which leads to poor predictive performance in most data experiments. After some reflection, this is because the baseline intensity part of the intensity function is fixed to a constant parameter value μ. Therefore, we try to propose alternative Hawkes process-based SRMs, so that the base strength becomes a time-dependent function. For the alternative exponential kernel HKP-based SRMs, the conditional intensity function takes the form as

$$\lambda(t) = \mu e^{-bt} + \sum_{T_i<t} e^{-bt}. \tag{12}$$

For the alternative pareto kernel HKP-based SRMs, we obtain

$$\lambda(t) = v\frac{bc^b}{(c+t)^{b+1}} + \sum_{T_i<t} \alpha \frac{bc^b}{(c+t)^{b+1}}. \tag{13}$$

We experiment with the alternative HKP-based SRMs using 8 data sets, where the experimental results obtained are shown in Table 5. Based on the results of Table 5, the alternative HKP-based SRMs with exponential and pareto kernels demonstrate notable improvements in their predictive performances in most scenarios, compared to the original HKP-based SRMs with simpler forms. Additionally, in half of the cases, the alternative HKP-based SRMs could exhibit predictive performance that is comparable to or even surpasses that of the NHPP-based SRMs, further highlighting their effectiveness in certain situations.

6 Conclusion

In this article, we proposed four software reliability models based on the simple Hawkes process and evaluated their predictive performances in comparison with traditional NHPP-based SRMs. The experimental results revealed that the predictive performance of the simple HKP-based SRMs did not outperform the NHPP-based SRMs. To address this limitation, we introduced an improvement by modifying the baseline intensity of the HKP-based SRMs, replacing the constant form with a time-dependent probability density function. This modification resulted in significant improvements in predictive performance, with the alternative HKP-based SRMs demonstrating comparable or even superior predictive accuracy to the NHPP-based SRMs in half of the cases.

These findings highlight the potential value of optimizing the baseline intensity in HKP-based SRMs for further research. However, this modification also increased the complexity of the models, making it more challenging to derive closed-form solutions of the mean value function. In future work, we plan to explore alternative approaches to obtain the mean value function, such as Monte Carlo simulation, to further enhance

Table 5: Predictive performance results by alternative HKP-based SRMs.

Dataset	Length	Model&Kernel	HKP AIC	HKP PMSE	NHPP AIC	NHPP PMSE	Alt. HKP AIC	Alt. HKP PMSE
TDS1	20%	Exp	**131.811**	88.781	141.609	**0.885**	143.609	**0.885**
		Pareto	145.750	100.301	**143.597**	**0.984**	145.607	2.064
	50%	Exp	**397.380**	46.655	403.368	**1.960**	405.368	**1.960**
		Pareto	412.396	48.111	405.241	**0.422**	407.259	4.916
	80%	Exp	703.454	10.410	**693.172**	1.562	694.932	**1.500**
		Pareto	720.037	10.003	**692.119**	**0.792**	693.511	2.378
TDS2	20%	Exp	**32.155**	3.428	41.394	**0.779**	43.394	3.489
		Pareto	45.379	6.727	**43.394**	**0.781**	45.480	3.130
	50%	Exp	**108.921**	3.704	123.477	1.200	126.409	**0.605**
		Pareto	127.456	4.076	**125.476**	**1.197**	127.491	4.194
	80%	Exp	148.356	1.483	180.401	1.487	176.000	**1.098**
		Pareto	184.396	2.904	182.401	**1.491**	184.556	2.816
TDS3	20%	Exp	168.505	23.178	**185.583**	**4.902**	189.583	4.926
		Pareto	191.583	28.013	189.583	4.931	191.767	**4.162**
	50%	Exp	428.761	63.906	446.961	**14.602**	448.960	14.630
		Pareto	450.960	66.998	448.960	14.620	452.725	21.042
	80%	Exp	765.838	5.450	**769.836**	1.516	771.836	1.516
		Pareto	790.317	5.259	770.823	**1.247**	771.475	2.826
TDS4	20%	Exp	296.611	160.580	304.679	16.509	**302.246**	**3.744**
		Pareto	308.678	168.496	**306.678**	16.548	305.156	56.494
	50%	Exp	759.235	213.845	773.710	26.408	**771.714**	**5.926**
		Pareto	777.708	212.041	**775.708**	**26.479**	760.653	34.194
	80%	Exp	1415.779	57.985	1392.340	**1.196**	**1362.000**	26.801
		Pareto	1430.865	65.843	1389.400	**0.713**	**1356.172**	13.006
TDS5	20%	Exp	**1024.168**	6.354	1046.920	0.414	1048.920	**0.412**
		Pareto	1050.923	6.861	**1048.920**	**0.412**	1050.928	7.134
	50%	Exp	**2580.375**	16.665	2602.910	**1.764**	2604.900	1.773
		Pareto	2606.905	16.540	**2604.900**	**1.771**	2593.773	55.710
	80%	Exp	**4180.704**	4.937	4203.050	**0.700**	4205.050	**0.700**
		Pareto	4207.462	5.046	4205.060	**0.699**	**4155.750**	23.136
TDS6	20%	Exp	**309.677**	214.801	313.856	**5.214**	321.319	224.576
		Pareto	321.319	225.236	**314.873**	**2.826**	315.856	5.215
	50%	Exp	866.581	108.194	**866.494**	**2.610**	871.833	93.968
		Pareto	880.385	112.026	**866.128**	**0.430**	868.494	2.610
	80%	Exp	1528.027	24.887	**1487.630**	**2.229**	1520.136	13.839
		Pareto	1542.242	24.564	**1481.450**	**1.125**	1489.960	1.813
TDS7	20%	Exp	**184.206**	144.529	191.227	**10.691**	193.227	10.769
		Pareto	195.229	134.113	**191.227**	10.573	195.294	**7.926**
	50%	Exp	538.462	75.412	**527.080**	**3.051**	529.080	**3.051**
		Pareto	543.787	78.307	**529.080**	**3.049**	532.717	3.336
	80%	Exp	946.271	11.670	**926.083**	1.406	927.372	**1.181**
		Pareto	957.726	11.136	**922.125**	**0.560**	924.226	5.799
TDS8	20%	Exp	**758.928**	635.488	764.584	**35.906**	764.886	395.179
		Pareto	768.582	642.936	766.582	36.059	**763.447**	**11.811**
	50%	Exp	1899.158	820.466	1904.770	**58.493**	**1881.202**	554.593
		Pareto	1908.763	824.218	1906.760	58.678	**1902.510**	**19.943**
	80%	Exp	3695.260	230.882	3489.310	**4.017**	**3420.119**	5.934
		Pareto	3706.932	230.906	3485.260	**2.613**	**3445.030**	4.876

the applicability and performance of the HKP-based SRMs.

Acknowledgments

This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University. This work was supported by JST SPRING, Grant Number JPMJSP2132.

References

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

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

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

[4] M. Xie. 1991. *Software Reliability Modeling*, World Scientific, Singapore.

[5] J. A. Achcar, D. K. Dey, and M. Niverthi. 1998. A Bayesian approach using non-homogeneous 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.

[6] H. Okamura and T. Dohi. 2013. SRATS: Software reliability assessment tool on spreadsheet, in *Proceedings of the 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE-2013)*, pp. 100–107, IEEE CPS.

[7] A. G. Hawkes., & D. Oakes. 1974. A cluster process representation of a self-exciting process. *Journal of applied probability*, 11(3), pp. 493-503.

[8] L. Lesage., M. Deaconu., A. Lejay., J. A. Meira., G. Nichil. and R. State. 2022. Hawkes processes framework with a Gamma density as excitation function: application to natural disasters for insurance. *Methodology and Computing in Applied Probability*, 24(4), pp. 2509-2537.

[9] N. Qiu., T. Dohi and H. Okamura. 2024. Evaluation of software reliability models based on Hawkes processes. In *Recent Advances in Reliability and Maintenance Modeling*, pp. 192-200. CRC Press.

[10] L. Cui., A. G. Hawkes., & H.Yi. 2020. An elementary derivation of moments of Hawkes processes. *Advances in Applied Probability*, 52(1), pp. 102-137.