

A Semi-Parametric Self-Exciting Software Reliability Model with Time-Varying Base Intensity and Shallow Neural Kernel

Nanxiang Qiu

Graduate School of Advanced Science and Engineering,
Hiroshima University, Hiroshima, Japan.
Corresponding author: d245541@hiroshima-u.ac.jp
ORCID: <https://orcid.org/0009-0004-1782-162X>

Tadashi Dohi

Graduate School of Advanced Science and Engineering,
Hiroshima University, Hiroshima, Japan.
E-mail: dohi@hiroshima-u.ac.jp
ORCID: <https://orcid.org/0000-0003-2954-0388>

Jingchi Wu

Graduate School of Advanced Science and Engineering,
Hiroshima University, Hiroshima, Japan.
E-mail: kure@hiroshima-u.ac.jp
ORCID: <https://orcid.org/0000-0001-8652-3002>

Junjun Zheng

Graduate School of Advanced Science and Engineering,
Hiroshima University, Hiroshima, Japan.
E-mail: jzheng@hiroshima-u.ac.jp
ORCID: <https://orcid.org/0000-0002-5529-1429>

Hiroyuki Okamura

Graduate School of Advanced Science and Engineering,
Hiroshima University, Hiroshima, Japan.
E-mail: okamu@hiroshima-u.ac.jp
ORCID: <https://orcid.org/0000-0001-6881-0593>

(Received on June 5, 2026; Revised on August 2, 2026; Accepted on August 30, 2026)

Abstract

Software reliability modeling is essential for assessing fault-detection behavior and predicting future software failures during the testing phase. Traditional nonhomogeneous Poisson process (NHPP)-based software reliability models (SRMs) are mathematically tractable, but they do not explicitly capture history-dependent fault-detection behavior. To address this limitation, this paper proposes a semi-parametric Hawkes process (HKP)-based SRM that combines an exponential time-varying base intensity with a shallow neural network (SNN) excitation kernel. The exponential base intensity represents the long-term decreasing trend of the background fault-detection rate, while the SNN excitation kernel flexibly learns the self-exciting effect induced by previous fault detections. To evaluate the expected cumulative number of detected faults, the mean value function is derived using an elementary approach for Hawkes processes and solved numerically. Experiments are conducted on eight real-world software fault-detection datasets. Our proposed semi-parametric SRM is compared with ten parametric HKP-based SRMs and representative NHPP-based SRMs using predictive mean squared error (PMSE). The results show that our proposed SRM achieves strong predictive performance, especially in early- and middle-stage prediction, and outperforms most competing SRMs in many settings. These findings indicate that the SNN excitation kernel provides a flexible and practically useful alternative for software reliability prediction.

Keywords- Software reliability model, Hawkes process, Self-excitation, Shallow neural network, Predictive performance.

1. Introduction

Reliability constitutes a fundamental quality concern for modern software systems. Their growing use in safety-critical, commercial, and infrastructure applications means that failures can result in service disruption, financial damage, and broader safety consequences. Consequently, an accurate description of fault-discovery behavior during testing is needed to support reliability evaluation, release decisions, testing-resource planning, and post-release maintenance. A wide variety of stochastic software reliability models (SRMs) have therefore been developed to characterize accumulated fault discoveries and forecast their future occurrence (Lyu, 1996; Musa et al., 1987; Xie, 1991).

NHPP-based SRMs remain one of the most established classes of reliability models because they combine a manageable mathematical structure with interpretable and practically useful predictions (Abdel-Ghaly et al., 1986; Achcar et al., 1998; Erto et al., 2020; Goel, 1985; Goel & Okumoto, 1979; Gokhale & Trivedi, 1998; Littlewood, 1984; Musa & Okumoto, 1984; Ohba, 1984; Ohishi et al., 2009; Okamura & Dohi, 2013; Zhao & Xie, 1996). Under this framework, fault discoveries are represented by a counting process whose expected trajectory coincides with the integrated intensity. Alternative specifications of the mean-value curve, or equivalently of the fault-detection-time distribution, lead to exponential, gamma, Pareto, log-normal, logistic, log-logistic, extreme-value, and Weibull-type SRMs. Together, these alternatives provide different descriptions of the overall accumulation pattern of detected faults throughout software testing.

However, NHPP-based SRMs usually rely on the assumption that the increments of the fault-count process are independent. Under this assumption, future fault detections are governed mainly by time-dependent deterministic intensity functions, and the direct influence of previous fault detections on subsequent detections is not explicitly modeled. This assumption may be too restrictive for practical software systems. In real testing environments, the detection of one fault may lead testers to inspect related modules, trigger additional test cases, or reveal dependent faults in connected components. As a result, software fault detections may occur in clusters rather than independently. This kind of temporal dependence is especially relevant in modern complex software systems, where system modules, execution contexts, and user interactions are highly interdependent.

To describe such history-dependent behavior, self-exciting point processes provide a natural modeling framework. In particular, the Hawkes process (HKP) (Hawkes, 1971; Hawkes & Oakes, 1974) is a representative self-exciting point process in which each event temporarily increases the probability of future events. Its conditional intensity consists of two parts: a base intensity that represents the background event-occurrence tendency, and an excitation term that accumulates the influence of previous events through an impact function, also called a self-excitation kernel. This structure is suitable for software reliability modeling because it can represent both the basic fault-detection trend and the local clustering effect caused by previously detected faults. Earlier studies have shown that self-exciting point processes can provide a unified view of software reliability modeling (Chen & Singpurwalla, 1997), and recent studies have further investigated HKP-based SRMs for software fault-count data (Iwamoto et al., 2022; Qiu et al., 2024; Qiu et al., 2025; Zhang et al., 2020).

Hawkes processes have been employed in a variety of areas, including finance, social-media analysis, and crime modeling (Lallouache & Challet, 2016; Lindstrom et al., 2023; Reinhart & Greenhouse, 2018; Rizoiu et al., 2022; Wang, 2015). Nevertheless, the classical formulation uses a time-invariant base intensity. Such an assumption is difficult to reconcile with software reliability growth, because the background tendency to discover faults generally diminishes as testing proceeds and detected faults are removed. A constant baseline may consequently exaggerate the late-stage discovery rate and inadequately represent the deceleration or saturation commonly found in reliability data. HKP-based SRMs with time-dependent base

intensities have therefore been investigated (Qiu et al., 2025; Zhang et al., 2020). Within these models, the baseline describes the global reliability-growth tendency, whereas the excitation component accounts for local dependence among fault discoveries.

Another important component of an HKP-based SRM is the excitation kernel. Previous HKP-based SRMs have often used parametric kernels, such as exponential, gamma, Pareto, log-normal, log-logistic, extreme-value, and Weibull-type kernels. These kernels are interpretable and computationally convenient, and they can describe different shapes of excitation decay after each fault detection. However, their predictive performance may vary considerably from dataset to dataset. For example, some kernels assume an immediately decreasing effect, whereas others assume a delayed peak or a particular tail-decay pattern. If the selected parametric shape differs substantially from the actual fault-detection dynamics, this shape misspecification may reduce predictive accuracy. Moreover, the most suitable parametric kernel may vary across software projects and observation stages. In practice, several candidate HKP-based SRMs must therefore be fitted and compared before an appropriate kernel can be selected. However, the true excitation pattern governing future fault detections is unknown at the time of model selection, and a kernel that fits the observed data well does not necessarily provide the best prediction for the unobserved testing period. This dataset- and stage-dependent model-selection problem motivates the use of a more flexible excitation kernel that can learn its shape directly from the observed data without requiring a single parametric family to be specified in advance.

In this paper, we propose a semi-parametric Hawkes process-based software reliability model by combining an exponential base intensity function with a shallow neural network (SNN) excitation kernel (Joseph et al., 2022). The model is called semi-parametric because its base intensity is specified by a parametric exponential form, while its excitation kernel is represented by a shallow neural network. The exponential base intensity describes the long-term decreasing tendency of the background fault-detection rate, whereas the SNN kernel flexibly characterizes the history-dependent excitation effect caused by previous fault detections. Compared with conventional HKP-based SRMs with fixed parametric kernels, our proposed semi-parametric SRM can represent more complex excitation patterns while preserving the explicit kernel-based structure of the Hawkes process. The SNN kernel was selected as a balance among flexibility, interpretability, and computational tractability. While spline- or basis-function-based kernels and Gaussian-process-based approaches can also model flexible excitation shapes, they typically require additional design choices (e.g., basis selection, knot placement, or inference procedures), which increase modeling and computational complexity in repeated reliability estimation. Deep neural temporal point-process models, such as LSTM-based Hawkes processes, can capture complex temporal dependencies but rely on latent hidden states, making individual excitation effects less directly interpretable and complicating long-term mean-value prediction. In contrast, the proposed SNN directly maps elapsed time since a past fault to its excitation contribution, preserving the explicit Hawkes structure. Its ReLU–exponential form further yields a piecewise exponential kernel with analytically tractable integrals, enabling efficient likelihood evaluation and mean-value computation. Unlike neural Hawkes models that represent the overall conditional intensity through latent recurrent states (Mei & Eisner, 2017), our model retains an explicit additive excitation kernel and applies the neural network only to learn its shape. The SNN kernel is combined here with a time-varying base intensity and adapted specifically to software reliability growth phenomena. A more detailed comparison with existing neural and semi-parametric Hawkes models is provided in Section 3.3.

Prediction with an HKP-based SRM additionally requires the expected trajectory of the cumulative fault count. Unlike the NHPP case, this quantity cannot be obtained by integrating the conditional intensity along a single observed path, because the intensity is random and depends on the preceding event history. We therefore apply the elementary moment argument of Cui et al. (2020) to derive the equation governing the

mean value function of the proposed semi-parametric SRM. Its numerical evaluation then provides the expected cumulative fault counts used in the predictive assessment.

The main objective of this paper is to evaluate the practical effectiveness of our proposed semi-parametric SRM from two perspectives. First, we investigate to what extent the SNN excitation kernel can serve as an alternative to conventional parametric excitation kernels in HKP-based SRMs. For this purpose, our proposed semi-parametric SRM is compared with HKP-based SRMs using ten representative parametric excitation kernels in terms of predictive performance. Second, we examine whether our proposed semi-parametric SRM is practically useful when compared with traditional NHPP-based SRMs, which remain one of the most widely used classes of software reliability models. Through this comparison, we aim to clarify whether our proposed SRM can provide competitive or improved reliability prediction performance against both self-exciting parametric-kernel SRMs and conventional NHPP-based SRMs.

The subsequent sections are arranged as follows. Section 2 introduces the stochastic foundations of NHPP-based and HKP-based reliability models, emphasizing their Markovian and history-dependent characteristics. Section 3 formulates the parametric-kernel HKP models and the proposed semi-parametric model with an exponential baseline and SNN excitation kernel. The derivation and numerical evaluation of the corresponding mean value function are presented in Section 4. Section 5 describes the estimation procedure, datasets, evaluation criteria, and comparative findings. Section 6 summarizes the main conclusions and discusses directions for further investigation.

2. Markovian and Non-Markovian Software Reliability Models

In this paper, the sequence of fault discoveries is represented by a stochastic counting process. Let $\{N(t), t \geq 0\}$ denote the number of faults found over the interval $(0, t]$. The modeling frameworks considered below differ mainly in the information used to determine the instantaneous fault-detection rate: a Markovian formulation is driven by the current time or state, whereas a non-Markovian formulation may additionally depend on the preceding event history.

2.1 NHPP-Based Software Reliability Models

A nonhomogeneous Poisson process (NHPP)-based SRM provides a Markovian description of the fault-count process. Counts accumulated over disjoint testing intervals are assumed to be independent, and their distribution is specified through a nonnegative deterministic rate $\lambda(t; \theta)$, where θ collects the free model parameters.

Let $P_n(t) = \Pr\{N(t) = n\}$ be the probability that n software faults have been detected by time t . The NHPP is characterized by the following Kolmogorov forward equations:

$$\frac{dP_0(t)}{dt} + \lambda(t; \theta)P_0(t) = 0 \quad (1)$$

$$\frac{dP_n(t)}{dt} = \lambda(t; \theta)\{P_{n-1}(t) - P_n(t)\}, \quad n \geq 1 \quad (2)$$

Subject to $P_0(0) = 1$ and $P_n(0) = 0$ for $n \geq 1$, the resulting state distribution is

$$P_n(t) = \frac{\{\Lambda(t; \theta)\}^n}{n!} e^{-\Lambda(t; \theta)}, \quad n = 0, 1, 2, \dots \quad (3)$$

here, the integrated rate is defined by

$$\Lambda(t; \theta) = \int_0^t \lambda(u; \theta) du \quad (4)$$

Consequently, the expected fault count coincides with the integrated rate,

$$m(t) = E[N(t)] = \Lambda(t; \theta) \quad (5)$$

This identity makes the NHPP framework mathematically tractable and convenient for reliability analysis. Once the parametric form of $\Lambda(t; \theta)$ is specified, both reliability assessment and future fault prediction can be performed straightforwardly.

A common way to construct NHPP-based SRMs is to express the mean value function as

$$m(t; \theta) = \omega F(t; \pi) \quad (6)$$

In this expression, $\omega > 0$ gives the limiting expected fault content, while $F(t; \pi)$ denotes a fault-detection-time distribution indexed by π . Selecting a different distribution for F produces a different reliability-growth curve. The specifications adopted as NHPP reference models are collected in **Table 1** (Okamura et al., 2013; Okamura & Dohi, 2013).

Table 1. NHPP reference specifications based on alternative fault-detection-time distributions.

NHPP specification	Fault-time distribution and mean value
Exponential (exp)	$m(t) = \omega F(t), F(t) = 1 - e^{-bt}$
Gamma (gamma)	$m(t) = \omega F(t), F(t) = \int_0^t \frac{c^b s^{b-1} e^{-cs}}{\Gamma(b)} ds$
Pareto (pareto)	$m(t) = \omega F(t), F(t) = 1 - (\frac{b}{t+b})^c$
Truncated normal (tnorm)	$m(t) = \omega \frac{F(t)-F(0)}{1-F(0)}, F(t) = \frac{1}{\sqrt{2\pi}b} \int_{-\infty}^t e^{-\frac{(s-c)^2}{2b^2}} ds$
Log-normal (lnorm)	$m(t) = \omega F(\ln t), F(t) = \frac{1}{\sqrt{2\pi}b} \int_{-\infty}^t e^{-\frac{(s-c)^2}{2b^2}} ds$
Truncated logistic (tlogist)	$m(t) = \omega \frac{F(t)-F(0)}{1-F(0)}, F(t) = \frac{1}{1+e^{-\frac{t-c}{b}}}$
Log-logistic (llogist)	$m(t) = \omega F(\ln t), F(t) = \frac{1}{1+e^{-\frac{t-c}{b}}}$
Truncated extreme-value max (txvmax)	$m(t) = \omega \frac{F(t)-F(0)}{1-F(0)}, F(t) = e^{-e^{-\frac{t-c}{b}}}$
Log-extreme-value max (lxvmax)	$m(t) = \omega F(\ln t), F(t) = e^{-e^{-\frac{t-c}{b}}}$
Truncated extreme-value min (txvmin)	$m(t) = \omega \frac{F(0)-F(-t)}{F(0)}, F(t) = e^{-e^{-\frac{t-c}{b}}}$
Log-extreme-value min (lxvmin)	$m(t) = \omega(1 - F(-\ln t)), F(t) = e^{-e^{-\frac{t-c}{b}}}$

$\Gamma(\cdot)$: Euler's gamma function

These models correspond to commonly used fault-detection time distributions and have been widely adopted in software reliability analysis. For every proper distribution listed in the table, $F(t; \pi)$ approaches one as the testing time increases. Hence,

$$\lim_{t \rightarrow \infty} m(t; \theta) = \omega \quad (7)$$

Thus, ω serves as the finite asymptotic fault content under the corresponding NHPP-based SRM.

Suppose that the ordered fault-discovery times observed up to T are $0 < t_1 < \dots < t_n \leq T$. The likelihood function of an NHPP-based SRM is

$$L(\theta) = e^{-\Lambda(T; \theta)} \prod_{i=1}^n \lambda(t_i; \theta) \quad (8)$$

Parameter estimation may therefore be carried out by maximizing $L(\theta)$, which is equivalent to minimizing $-\log L(\theta)$.

Although NHPP-based SRMs have been widely used, they have an important limitation. Since the intensity function $\lambda(t; \theta)$ is deterministic and depends only on time, the model does not explicitly describe the influence of previous fault detections on future fault detections. In other words, the occurrence of a fault at time t_i does not directly change the future intensity except through the pre-specified time-dependent trend. This assumption may be restrictive for complex software systems, where one detected fault may lead to additional inspections, reveal related faults, or trigger fault detections in dependent modules.

2.2 Classical Hawkes Process

Historical feedback can be introduced through a Hawkes process (HKP) (Hawkes, 1971; Hawkes & Oakes, 1974). In this point-process formulation, the occurrence of an event modifies the short-term rate of later events. For software testing, this mechanism allows a newly discovered fault to influence the likelihood of finding related faults subsequently.

Write $\mathcal{H}_t\{t_i \mid t_i < t\}$ for the fault-detection history available immediately before t . The classical univariate HKP assigns the following conditional rate:

$$\lambda(t \mid \mathcal{H}_t) = \mu + \int_0^t \phi(t - u) dN(u) = \mu + \sum_{t_i < t} \phi(t - t_i) \quad (9)$$

Here, $\mu > 0$ specifies the event rate in the absence of historical excitation, and $\phi(\tau)$ measures the additional contribution of an event after a lag τ . The summation therefore superposes the residual effects of all faults detected before t .

The HKP has several important properties that make it useful for software reliability modeling. First, it has a self-exciting property. A positive kernel produces endogenous excitation. Specifically, $\phi(\tau) > 0$ means that a fault detected at t_i contributes a positive amount to the rate at $t_i + \tau$. For $\phi(\tau) = \alpha\beta e^{-\beta\tau}$, this contribution is largest immediately after the detection and diminishes continuously with the elapsed time. Consequently, multiple fault discoveries may be concentrated within relatively short testing intervals.

Second, the HKP is generally non-Markovian. Given that the instantaneous rate is contingent upon the complete filtration $\mathcal{H}_t$, predicting the system's trajectory requires more than just the present time and accumulated faults. Instead, the timing and spacing of previous fault detections affect the next-event distribution. This is important in software testing because closely spaced fault detections may indicate a temporary increase in testing activity, fault dependency, or local instability in related software modules.

Third, the process also admits a cluster interpretation. Events attributed to the baseline act as initial discoveries, and each such event may initiate further detections according to ϕ . Descendants can themselves generate additional events, providing a probabilistic representation of fault clusters that propagate across related modules or testing scenarios.

2.3 Relationship Between NHPP-Based and HKP-Based SRMs

The distinction between the two formulations lies in the argument of the event rate. An NHPP uses an intensity determined by the absolute testing time, whereas an HKP adds contributions indexed by the lags $t - t_i$ from earlier discoveries. If the history-dependent contribution is removed, the conditional intensity becomes deterministic and the usual NHPP formulation is recovered.

For example, consider the exponential-type excitation kernel in an HKP-based SRM $\phi(t - t_i) = be^{-b(t-t_i)}$, if the kernel function becomes dependent only on t and not on the time of past events, it becomes be^{-bt} , by integrating this deterministic intensity over $(0, t]$, we obtain

$$\Lambda(t) = \int_0^t b e^{-bs} ds = 1 - e^{-bt} \quad (10)$$

which is the mean value function of the exponential NHPP-based SRM. This example shows that the HKP-based formulation extends the NHPP-based formulation by shifting the density function from the absolute testing time t to the elapsed time $t - t_i$ after each previous fault detection.

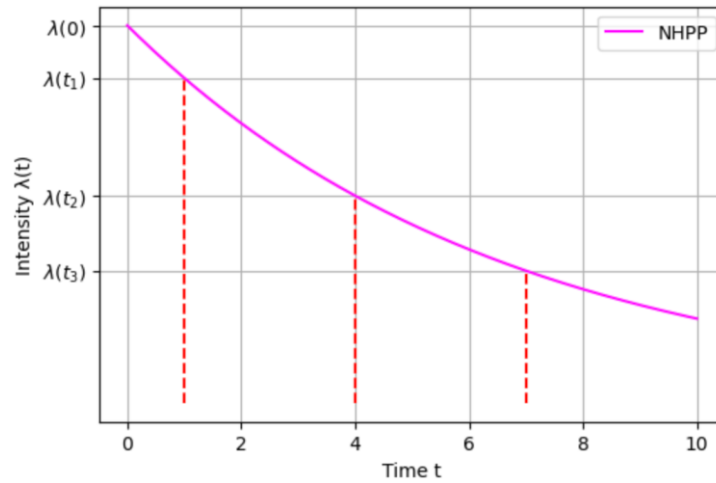


Figure 1. Typical intensity function of NHPP-based and HKP-based SRMs.

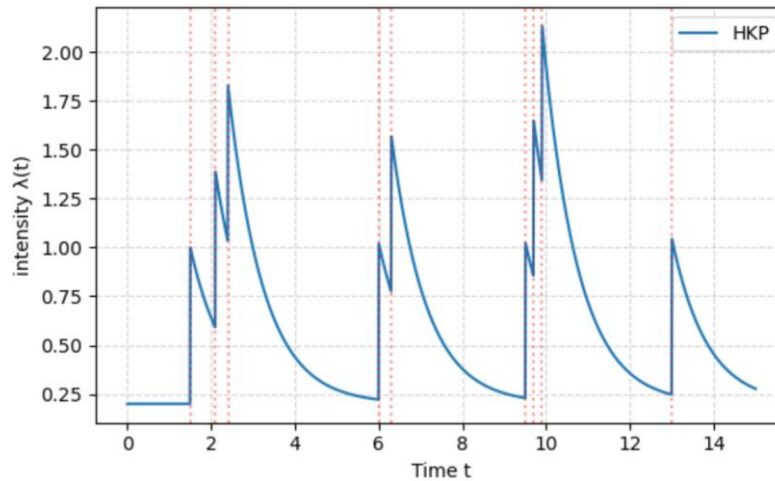


Figure 2. Typical intensity function of HKP-based SRMs.

The difference can be understood visually from the shapes of the intensity functions. **Figures 1 and 2** compare the typical intensity profiles of NHPP-based and HKP-based SRMs. As shown in **Figure 1**, the NHPP intensity is a smooth deterministic curve driven only by the testing time. It captures the global decreasing trend of fault detection but does not react to individual fault-detection events. In contrast, **Figure 2** shows that the HKP intensity increases immediately after each fault detection and then gradually decays according to the excitation kernel. This spike-and-decay pattern reflects the self-exciting mechanism of the

HKP, in which each detected fault may temporarily increase the likelihood of subsequent fault detections. To further support HKP modeling motivation, we can examine the empirical intensity pattern of real software fault-detection data. Since the intensity represents the local rate of event occurrence, a simple first-order approximation is given by the reciprocal of the inter-arrival time:

$$\frac{1}{\Delta t_i} = \frac{1}{t_i - t_{i-1}} \quad (11)$$

This quantity can be regarded as a naive pseudo-intensity at each fault-detection time t_i . A shorter inter-arrival time indicates a higher local fault-detection rate and therefore produces a higher pseudo-intensity peak.

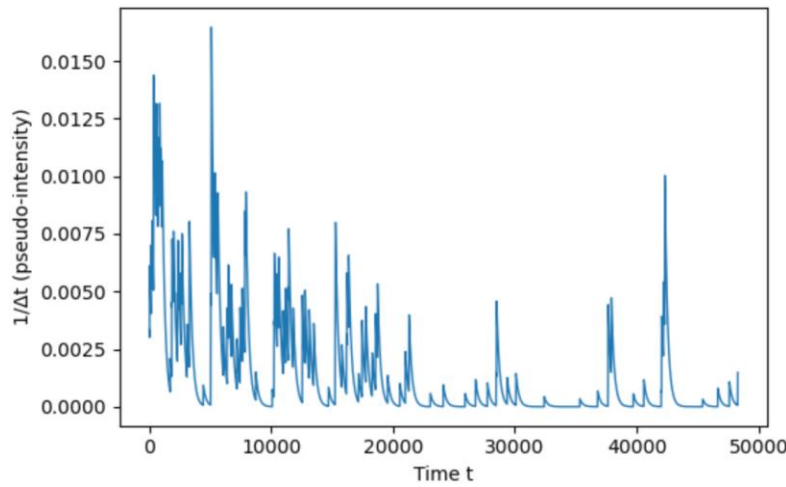


Figure 3. Naive pseudo-intensity for a real-world software fault-detection dataset.

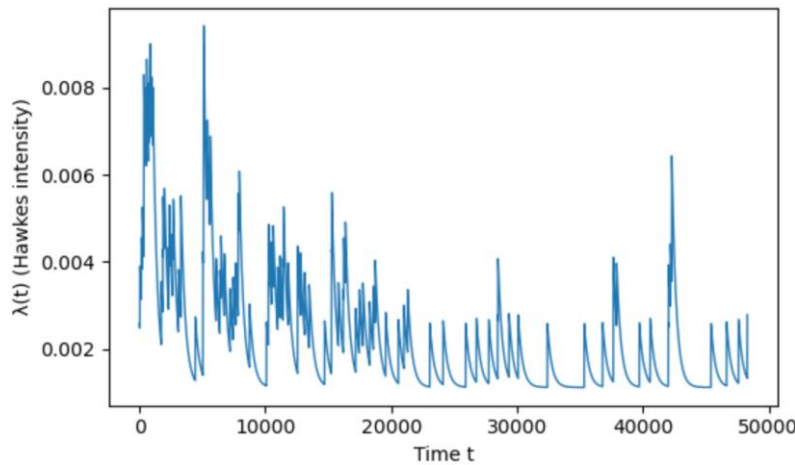


Figure 4. Fitted HKP conditional intensity for the same fault sequence.

Figure 3 plots the naive pseudo-intensity for a real-world software fault-detection dataset. The empirical curve is highly irregular, clustered, and spike-like, suggesting that the local fault-detection rate changes

sharply over time. **Figure 4** shows the fitted HKP intensity for the same dataset. The fitted HKP intensity exhibits a similar jagged and decaying-spike pattern, which is much closer to the empirical pseudo-intensity than a smooth NHPP intensity curve. This visual correspondence provides an intuitive justification for using the HKP framework to model software fault-detection processes with temporal dependence and self-excitation.

Overall, the HKP-based SRM provides a more general framework that can represent the local self-exciting behavior observed in software fault-detection data.

3. HKP-Based SRMs with Exponential Base Intensity

Section 2 showed that the HKP-based framework extends NHPP-based SRMs by incorporating history-dependent self-excitation into the fault-detection process. In software reliability modeling, however, a constant background rate is not sufficient (Qiu et al., 2025; Zhang et al., 2020), because the background fault-detection tendency usually decreases as testing progresses and faults are detected and removed. Therefore, we adopt a time-varying base intensity to describe the long-term reliability growth trend.

Given the pre- t fault history $\mathcal{H}_t\{t_i \mid t_i < t\}$, the conditional rate of the resulting reliability model is written as

$$\lambda(t \mid \mathcal{H}_t) = \mu(t) + \int_0^t \phi(t-s) dN(s) = \mu(t) + \sum_{t_i < t} \phi(t-t_i) \quad (12)$$

Here, $\mu(t)$ describes the time-dependent background rate, whereas $\phi(\tau)$ quantifies the residual excitation produced by a fault after an elapsed time τ . Specifically,

$$\mu(t) = vbe^{-bt}, \quad v > 0, b > 0 \quad (13)$$

the parameter v determines the overall magnitude of the baseline, while b governs its rate of decay. Because $\mu(t)$ decreases monotonically, this specification reflects a background fault-discovery tendency that weakens as testing advances.

The exponential form is adopted as the starting point of the present semi-parametric modeling framework for both theoretical and practical reasons. It is one of the simplest and most widely used fault-detection-rate forms in software reliability modeling. Its monotonically decreasing shape represents the gradual reduction in spontaneous fault detection as testing progresses, while its small number of parameters provides clear interpretation and stable estimation. In addition, its cumulative base intensity is analytically tractable.

3.1 Parametric HKP-Based SRMs

We have proposed the purely parametric HKP-based SRMs (Qiu et al., 2024; Qiu et al., 2025) constructed by combining the exponential time-varying base intensity in Equation (13) with parametric excitation kernels. In these models, the excitation kernel is specified by a pre-defined parametric density-type function. Let $\tau = t - t_i$ denote the elapsed time since a previous fault detection at time t_i . A parametric excitation kernel can generally be written as

$$\phi(\tau; \pi) = \alpha f(\tau; \eta) \quad (14)$$

where, $f(\tau; \eta)$ denotes a density-type function, η is its free parameter vector, and $\alpha > 0$ controls the strength of self-excitation. A larger α means that each previous fault detection has a stronger influence on subsequent fault detections.

By using different forms of $\phi(\tau; \pi)$, we can represent different excitation patterns after a fault is detected. For example, the exponential kernel describes an excitation effect that is strongest immediately after a fault detection and then decreases monotonically. In contrast, the gamma or log-normal kernels can describe delayed excitation effects, where the influence first increases to a peak and then decreases. Such flexibility is useful because different software projects may exhibit different fault-dependency patterns.

Table 2 summarizes the ten representative parametric excitation kernels considered in this paper. These kernels correspond to commonly used distributional forms in software reliability modeling and are used to construct ten parametric HKP-based SRMs.

Table 2. Representative parametric excitation kernels in HKP-based SRMs.

Kernel family	Excitation function $\phi(\tau; \pi)$
Exponential (exp)	$\frac{abe^{-b\tau}}{c}$
Gamma (gamma)	$\frac{\alpha c^b \tau^{b-1} e^{-c\tau}}{\Gamma(b)}$
Pareto (pareto)	$\frac{abc}{(c+\tau)^2} \left(\frac{c}{c+\tau}\right)^{b-1}$
Log-normal (lnorm)	$\frac{\alpha e^{-\frac{(c-\log \tau)^2}{2b^2}}}{b\sqrt{2\pi\tau}}$
Truncated logistic (tlogist)	$\frac{b(1 + e^{-(\tau-c)/b})^2 (1 - \frac{1}{1 + e^{c/b}})}{\alpha e^{-(\log \tau - c)/b}}$
Log-logistic (llogist)	$\frac{b(1 + e^{-(\log \tau - c)/b})^2 \tau}{\alpha e^{-\frac{\tau-c}{b} - (\tau-c)/b}}$
Truncated extreme-value max (txvmax)	$\frac{b(1 - e^{-e^{c/b}})}{\alpha e^{-e^{-\frac{\tau-c}{b}} - (\tau-c)/b}}$
Log-extreme-value max (lxvmax)	$\frac{b}{\alpha e^{-e^{-\frac{c-\log \tau}{b}} + (c-\log \tau)/b}}$
Truncated extreme-value min (txvmin)	$\frac{h}{\alpha e^{\frac{c}{b} - e^{\frac{c+\tau}{b}} + \frac{(\tau+c)}{b}}}$
Weibull	$\frac{b}{\alpha} \left(\frac{\tau}{c}\right)^{b-1} e^{-(\tau/c)^b}$

$\Gamma(\cdot)$: Euler's gamma function.

The advantage of these parametric HKP-based SRMs is that each kernel has a clear distributional interpretation. The model can describe both the long-term decreasing trend of software fault detection through $\mu(t)$, and the local clustering of fault detections through the ten excitation kernels $\phi(\tau; \pi)$. However, their flexibility is limited by the pre-specified functional form of the kernel. In practical datasets, the most suitable kernel may vary from one software project to another. As also demonstrated by the comparative results in Section 5, no single parametric kernel consistently provides the best prediction across all datasets and observation points. Consequently, relying on parametric kernels requires repeated fitting and selection among multiple candidate models. This motivates the introduction of a more flexible excitation kernel.

3.2 Semi-Parametric Model with SNN Excitation Kernel

To overcome the limitation of fixed parametric excitation kernels, we introduce a shallow neural network (SNN) (Joseph et al., 2022) excitation kernel into the HKP-based SRM with exponential time-varying base intensity. Our proposed model is referred to as the semi-parametric SRM in this paper. It is called semi-parametric because the base intensity is still specified by the parametric exponential form in Equation (13),

while the excitation kernel is represented by an SNN and is therefore more flexible than a fixed parametric kernel. The choice of a shallow rather than deep neural architecture is deliberate. Our purpose is not to replace the complete conditional intensity with a black-box sequence model, but to learn only the shape of the excitation kernel while retaining the additive Hawkes structure. This shallow formulation provides sufficient flexibility for the one-dimensional elapsed-time input, requires fewer parameters than deep recurrent architectures, and preserves the explicit kernel form needed for likelihood calculation and mean-value prediction.

We parameterize the excitation function by a single-hidden-layer feed-forward network. Denote its hidden width by H and use the event lag $\tau \geq 0$ as the scalar input. The resulting kernel takes the form

$$\phi_{SNN}(\tau; \Theta) = \exp(b_2 + \sum_{j=1}^H a_{2,j} \max(a_{1,j}\tau + b_{1,j}, 0)) \quad (15)$$

where, $\Theta = (b_2, \{a_{1,j}, a_{2,j}, b_{1,j}\}_{j=1}^H)$ is the parameter vector of the SNN kernel. The ReLU activation function $\max(a_{1,j}\tau + b_{1,j}, 0)$ allows the kernel shape to change piecewise over time, while the exponential output function ensures that the kernel value remains positive.

Substitution of the exponential baseline and the learned SNN kernel into Equation (12) yields the proposed semi-parametric intensity:

$$\lambda_{SP}(t | \mathcal{H}_t) = vbe^{-bt} + \sum_{t_i < t} \phi_{SNN}(t - t_i; \Theta) \quad (16)$$

where, the parameter vector of the proposed model is $\theta_{SP} = (v, b, \Theta)$.

Although the excitation kernel is represented by a neural network, it still has a clear interpretation in the HKP framework. The input τ represents the elapsed time after a past fault detection, and the output $\phi_{SNN}(\tau)$ represents the additional fault-detection intensity induced by that past fault after time τ . Therefore, the SNN kernel does not replace the Hawkes process with a black-box temporal model. Instead, it only replaces the fixed parametric kernel with a data-adaptive kernel while preserving the explicit self-exciting structure. The learned kernel can still be plotted and interpreted as the time-varying influence of a previously detected fault.

The structure of Equation (15) also explains why the proposed SNN kernel is computationally tractable. Each hidden neuron has the form $\max(a_{1,j}\tau + b_{1,j}, 0)$, ReLU unit changes its activation status only at the breakpoint $\tau_j = -b_{1,j}/a_{1,j}$. After collecting and sorting all positive breakpoints, the time axis can be divided into several intervals. On each interval, the set of active ReLU units is fixed. Let $I_k = [\tau_k, \tau_{k+1})$ denote one such interval, and let J_k be the set of active hidden neurons on I_k . Then, for $\tau \in I_k$, Equation (15) becomes

$$\phi_{SNN}(\tau; \Theta) = \exp(b_2 + \sum_{j \in J_k} a_{2,j}(a_{1,j}\tau + b_{1,j})) = \exp(c_k + d_k\tau), \tau \in I_k \quad (17)$$

where, $c_k = b_2 + \sum_{j \in J_k} a_{2,j}b_{1,j}$, $d_k = \sum_{j \in J_k} a_{2,j}a_{1,j}$.

Hence, once the active ReLU set is fixed, the neural kernel reduces locally to an exponential function. Its integral can consequently be evaluated exactly within each activation interval. For any $\alpha, \gamma \in I_k$ with $\alpha < \gamma$, we have

$$\int_{\alpha}^{\gamma} \phi_{SNN}(\tau; \Theta) d\tau = \begin{cases} \frac{e^{c_k}}{d_k} (e^{d_k\gamma} - e^{d_k\alpha}), & d_k \neq 0, \\ e^{c_k}(\gamma - \alpha), & d_k = 0. \end{cases} \quad (18)$$

If the integration interval crosses multiple breakpoints, the integral can be obtained by splitting the interval at those breakpoints and summing the corresponding closed-form integrals. Thus, in the log-likelihood function of an HKP-based SRM, the integral of the self-excitation term over the observation period can be transformed as

$$\int_0^T \sum_{t_i < u} \phi_{SNN}(u - t_i; \Theta) du = \sum_{i=1}^n \int_{t_i}^T \phi_{SNN}(u - t_i; \Theta) du \quad (19)$$

Therefore, the integral term in the HKP likelihood can be evaluated efficiently using the piecewise closed-form expression in Equation (18). This is an important advantage of the proposed SNN kernel: it improves the flexibility of the excitation function while keeping the likelihood computation tractable.

3.3 Relationship to Existing Neural and Semi-Parametric Hawkes Models

Representative neural Hawkes models use continuous-time recurrent states to encode the complete event history and parameterize the conditional intensity (Mei & Eisner, 2017). These models can represent complex temporal dependencies, but their hidden-state or attention-based formulations generally do not retain an explicit additive excitation kernel that directly describes the effect of each individual previous event. Their relatively large parameter spaces may also be unnecessarily complex for the small univariate software fault datasets considered in this paper. In contrast, the proposed model preserves the conventional additive Hawkes structure and uses the neural network only to represent the one-dimensional relationship between elapsed time and additional excitation intensity. Consequently, the learned excitation effect remains explicit, plottable, and directly interpretable.

The SNN kernel used in this paper follows (Joseph et al., 2022), whose principal objective was to provide flexible non-parametric estimation of Hawkes excitation kernels under a time-invariant base intensity. The present paper differs mainly in its software reliability formulation and prediction objective. We combine the SNN kernel with an exponential time-varying base intensity to represent the decreasing background fault-detection tendency during software testing. Moreover, the present study proceeds beyond excitation-kernel estimation and short-window event forecasting. A Volterra equation is established for the expected fault-count trajectory and is used to extrapolate fault accumulation throughout the unobserved portion of testing. The resulting framework links an explicit data-adaptive excitation mechanism to reliability-growth analysis and extended-horizon fault-count forecasting.

4. Expected Fault-Count Dynamics of HKP-Based SRMs

Software reliability prediction requires the expected trajectory of the accumulated fault count. For an NHPP-based SRM, this trajectory equals the integrated deterministic intensity. The same calculation cannot be applied directly to an HKP-based SRM, because its conditional rate changes with the random sequence of previously observed events. Integrating that rate along one realized history would therefore describe only that particular path rather than the process expectation.

We obtain the required expectation using the elementary moment argument developed by Cui et al. (2020). Their approach relates infinitesimal changes in a counting process to the expectation of its conditional intensity and can be applied to both Markovian and selected non-Markovian Hawkes models.

4.1 Basic Idea of the Elementary Approach

Define $m(t) = E[N(t)]$. Conditional on the available history $\mathcal{H}_t$, the expected number of events occurring during a short interval of length Δt satisfies

$$E[N(t + \Delta t) - N(t) | \mathcal{H}_t] = \lambda(t | \mathcal{H}_t) \Delta t + o(\Delta t) \quad (20)$$

Taking expectation on both sides and letting $\Delta t \rightarrow 0$, we obtain

$$m'(t) = E[\lambda(t | \mathcal{H}_t)] \quad (21)$$

For convenience, set $q(t) = m'(t)$. This equation is the key idea used in this paper. It shows that, for an HKP-based SRM, the derivative of the mean value function is not the conditional intensity itself, but the expectation of the conditional intensity. For the general HKP-based SRM, substituting Equation (12) into Equation (21) gives

$$q(t) = \mu(t) + E \left[\int_0^t \phi(t-s) dN(s) \right] \quad (22)$$

Since the expected increment of the counting process satisfies $E[dN(s)] = dm(s) = q(s)ds$, the expected excitation term becomes

$$E \left[\int_0^t \phi(t-s) dN(s) \right] = \int_0^t \phi(t-s) q(s) ds \quad (23)$$

Combining Equations (22) and (23) produces a linear Volterra convolution equation for the expected intensity:

$$q(t) = \mu(t) + \int_0^t \phi(t-s) q(s) ds, \quad m(t) = \int_0^t q(s) ds, \quad m(0) = 0 \quad (24)$$

Equation (24) is common to every HKP-based SRM examined below. A particular choice of baseline and excitation function determines $q(t)$, after which integration of q gives the expected cumulative fault count.

4.2 Application to the HKP-Based SRMs

For the parametric candidates in Section 3.1, all models share the exponential baseline but differ in the kernel selected from **Table 2**. Let $q_r(t) = m'_r(t)$ denote the expected intensity associated with the r -th kernel, $r = 1, \dots, 10$. Equation (24) then becomes

$$q_r(t) = vbe^{-bt} + \int_0^t \phi_r(t-s; \pi_r) q_r(s) ds, \quad m_r(t) = \int_0^t q_r(s) ds \quad (25)$$

where, $\phi_r(\cdot; \pi_r)$ denotes the r -th parametric excitation kernel.

For our proposed semi-parametric SRM, the exponential base intensity is retained, but the fixed parametric kernel is replaced by the SNN excitation kernel introduced in Section 3.2. Thus, the expected intensity $q_{SP}(t)$ satisfies

$$q_{SP}(t) = vbe^{-bt} + \int_0^t \phi_{SNN}(t-s; \Theta) q_{SP}(s) ds, \quad m_{SP}(t) = \int_0^t q_{SP}(s) ds \quad (26)$$

Equation (26) is the final mean value equation of the semi-parametric SRM. Compared with Equation (25), the only structural difference is the replacement of the fixed parametric kernel $\phi_r(\cdot)$ with the data-adaptive SNN kernel $\phi_{SNN}(\cdot; \Theta)$.

In this paper, we evaluate the mean value function numerically. Let $y(t) = m'(t)$. After setting a time grid over the observation interval, the integral term in the Volterra equation is approximated by numerical quadrature, and $y(t)$ is computed recursively from earlier time points to later time points. The mean value function is then obtained by numerically integrating $y(t)$ over time:

$$m(t) = \int_0^t y(s) ds \quad (27)$$

For the parametric HKP-based SRMs, the required integrated kernels are obtained from their corresponding parametric forms. For the semi-parametric SRM, the SNN kernel has the piecewise exponential structure

described in Section 3.2, so its integrated kernel can be evaluated by summing closed-form integrals over the corresponding intervals. Thus, both the parametric and semi-parametric HKP-based SRMs can be evaluated under the same numerical framework.

4.3 Numerical Procedure and Computational Accuracy

The elementary approach described in Section 4.1 is used to derive the Volterra integral equation, whereas the resulting equation is evaluated by numerical quadrature. To reduce numerical scale differences among the datasets, all event times are first normalized to the interval $[0,1]$.

For prediction, let T_{ob} denote the observation point and let $H_{T_{ob}}$ denote the observed fault-detection history. The conditional expected intensity after T_{ob} satisfies:

$$r(t) = g(t) + \int_{T_{ob}}^t \phi(t-s) r(s) ds \quad (28)$$

where, $g(t) = \mu(t) + \sum_{t_i \leq T_{ob}} \phi(t-t_i)$ contains the base intensity and the excitation contributions of the observed faults.

A uniform grid $T_{ob} = s_0 < s_1 < \dots < s_{M-1} = T_{max}$ is used. Letting $h = s_i - s_{i-1}$ and $r_i \approx r(s_i)$, the Volterra equation is evaluated recursively as:

$$r_i = g(s_i) + h \sum_{j=0}^{i-1} \phi(s_i - s_j) r_j, \quad i = 1, 2, \dots, M-1 \quad (29)$$

Because this is a simple Volterra equation, each value is obtained directly from previously computed grid values, and no additional fixed-point iteration is required. The predicted cumulative number of faults is then calculated by numerical accumulation of r_i , and linear interpolation is used to obtain predictions at the actual test-event times.

Numerical accuracy is controlled by the grid resolution. In practice, for a given dataset, the number of grid points M is increased progressively (e.g., starting from 1000 and increasing in steps of 1000). The computation is considered sufficiently accurate once the relative change in key quantities, such as the PMSE or the final predicted cumulative fault count, becomes negligible between successive refinements. Time normalization and the triangular forward-recursion structure also improve numerical stability and reduce scaling issues across datasets.

The computed trajectory provides the model-based forecast of fault accumulation. Section 5 evaluates this trajectory against the observed counts in the withheld portion of each dataset.

5. Experiments

The empirical analysis is organized around two comparisons. The first comparison keeps the exponential baseline and the additive Hawkes structure unchanged and varies only the representation of the excitation kernel. This setting allows the predictive contribution of the SNN kernel to be assessed against ten distribution-based kernels. The second comparison uses representative NHPP-based SRMs as Markovian benchmarks and examines whether incorporating event-history dependence improves fault-count forecasting.

5.1 Data Sets and Experimental Settings

Eight event-time sequences are used in the prediction experiments, as listed in **Table 3**. DS1–DS4 are benchmark fault-detection records taken from established software reliability studies, whereas DS5–DS8

were constructed from fault-related issue histories of four open-source projects hosted on GitHub. The datasets contain between 104 and 397 recorded faults and cover substantially different testing-time scales, thereby providing heterogeneous conditions for evaluating the candidate models.

Table 3. Software fault-detection time-domain datasets.

Dataset ID	Number of faults	Final recorded time	Data origin
DS1	197	50,236,822.00	SYS4 (Musa, 1979)
DS2	136	88,682.00	SYS1 (Musa, 1979)
DS3	104	15,369.50	SRC3 (Lyu, 1996)
DS4	397	108,890.00	SRC1 (Lyu, 1996)
DS5	278	173,951.12	GitHub (ChatGPT)
DS6	278	61,149.81	GitHub (MobX)
DS7	183	40,241.43	GitHub (XGBoost)
DS8	290	35,310.04	GitHub (uni-app)

Let $T_{\max}$ denote the final recorded time of a dataset. Three observation boundaries are considered, namely $T_{\text{ob}} = 0.2T_{\max}$, $0.5T_{\max}$, and $0.8T_{\max}$. They represent predictions made from relatively early, intermediate, and late testing information. For a given boundary, only events satisfying $t_i \leq T_{\text{ob}}$, enter parameter estimation and model selection, all later events are withheld and used exclusively for out-of-sample evaluation. Thus, every model is fitted and assessed using the same temporal split.

5.2 Parameter Estimation and Model Selection

Consider the ordered training events $\mathcal{D}_T = t_1, \dots, t_n$, observed before the fitting horizon T . For any candidate point-process model, parameter estimation is based on the event-time log-likelihood

$$\ell(\boldsymbol{\theta}; \mathcal{D}_T) = - \int_0^T \lambda(s | \mathcal{H}_s; \boldsymbol{\theta}) ds + \sum_{i=1}^n \log \lambda(t_i | \mathcal{H}_{t_i}; \boldsymbol{\theta}) \quad (30)$$

The vector $\boldsymbol{\theta}$ contains the free model coefficients. In an NHPP-based SRM, the history argument in Equation (30) is irrelevant because the rate is deterministic. For an HKP-based SRM, the same expression is evaluated conditionally on the fault detections preceding each event time.

The NHPP benchmarks and the ten distributional-kernel HKP models are fitted by maximizing Equation (30). Every positive coefficient is represented as $\theta_j = \exp(\eta_j)$ and optimization is performed with respect to the unconstrained variable η_j . Particle swarm optimization is used to explore the parameter space, after which L-BFGS-B refines the candidate solutions. Repeated initializations are employed to reduce the risk that the retained estimate is determined by an unfavorable starting point.

For parametric models, repeated fits and alternative parametric specifications are compared using the Akaike information criterion (AIC) which is defined as

$$\text{AIC} = 2k - 2\ell(\hat{\boldsymbol{\theta}}) \quad (31)$$

Here, k is the number of estimated coefficients and $\ell(\hat{\boldsymbol{\theta}})$ is the optimized log-likelihood. The retained parametric fit is the one attaining the lowest AIC, so that likelihood improvement is considered together with the dimension of the model. When a single NHPP benchmark is required in **Table 5**, the minimum-AIC NHPP specification is selected.

The SNN specification is handled separately because its effective dimension changes with the hidden-layer width and the selected hyperparameter configuration. Rather than comparing these architectures through AIC, candidate configurations are ranked by the training objective $\mathcal{L}(\boldsymbol{\theta}) = -\ell(\boldsymbol{\theta}; \mathcal{D}_T)$. The configuration

producing the smallest negative log-likelihood is subsequently used for prediction.

Optimization of the semi-parametric SRM is achieved through the minimization of the negative log-likelihood. During training, the Adam algorithm is utilized to compute the variables for the exponential baseline intensity, alongside the synaptic connections and biases within the SNN excitation kernel. To reduce the sensitivity of the non-convex optimization to initialization, the SNN model was trained five times from independently initialized neural-network weights for each dataset and observation point. Training stability was further supported by L2 regularization, gradient clipping, adaptive learning-rate reduction, and early stopping. As discussed in Section 3.2, the SNN kernel has a piecewise exponential structure, so its integrated kernel can be evaluated efficiently during likelihood calculation. Among the trained semi-parametric SRM under different hyperparameter settings, the one with the minimum loss value is selected for subsequent prediction.

5.3 Predictive Performance Metric

Forecast accuracy is assessed only on events occurring after T_{ob} . Let $\hat{m}(t)$ denote the cumulative fault count predicted from the fitted mean-value equation. Its discrepancy from the held-out trajectory is summarized by PMSE.

Suppose that the complete dataset contains m event times and that the first p events lie within the training interval. At the held-out event time t_i , the observed cumulative count is n_i and its model-based counterpart is $\hat{m}(t_i)$. We calculate

$$PMSE = \sqrt{\frac{\sum_{i=p+1}^m (E[N(t_i)] - n_i)^2}{m-p}} \quad (32)$$

where, $E[N(t_i)]$ denotes the expected total fault count at time t_i , derived directly from the mean value function of the specific model. A smaller PMSE value indicates better predictive accuracy.

5.4 Results and Discussion

Figures 5 to 7 illustrate the three forecasting regimes on DS2. With the observation boundary placed at $0.2T_{max}$, none of the three trajectories reproduces the entire held-out sequence accurately. Nevertheless, the semi-parametric and NHPP curves develop more gradually than the selected parametric HKP forecast, whose cumulative prediction begins to rise too rapidly near the start of the withheld interval. When half of the testing horizon is available, both Hawkes formulations improve substantially. The SNN-based trajectory remains closest to the withheld cumulative counts over most of the interval, whereas the NHPP benchmark retains a noticeable systematic deviation. At $0.8T_{max}$, the differences among the forecasts narrow. The selected parametric HKP and NHPP models approach the terminal count more closely, while the SNN-based forecast ends with a modest positive bias.

Table 4 summarizes the PMSE comparison between our proposed semi-parametric SRM and the ten parametric HKP-based SRMs with different excitation kernels. The purpose of this comparison is to examine whether the SNN excitation kernel can serve as a flexible alternative to fixed parametric kernels.

As shown in **Table 4**, the introduced semi-parametric SRM achieves strong predictive performance, especially at the 20% and 50% observation points. At the 20% observation point, our proposed SRM obtains the best PMSE values on five out of eight datasets. Similarly, at the 50% observation point, it also achieves the best PMSE values on five out of eight datasets. These results suggest that the flexibility of the semi-parametric SRM is particularly beneficial in early- and middle-stage prediction. In these stages, the

collected failure data remains limited, while the underlying distributional pattern of the excitation effect is not sufficiently clear. Therefore, a fixed parametric kernel might struggle to capture the actual defect-discovery dynamics well. By contrast, the SNN kernel can adapt its shape to the observed data and thus provides better predictive performance in many cases.

At the 80% observation point, our proposed semi-parametric SRM achieves the best PMSE value on two datasets. This does not mean that the semi-parametric SRM becomes ineffective in the late stage. Rather, the remaining data mainly correspond to the sparse late stage of software testing, where the self-exciting effect becomes weaker and the cumulative fault-detection trend is relatively smooth and easier to characterize using a fixed parametric form. The flexibility of the SNN kernel may also lead to mild overfitting in this setting. Since the model is selected according to the minimum training negative log-likelihood, it may fit local fluctuations and irregular excitation patterns over the longer observed history that may not persist in the remaining late-stage interval. In contrast, parametric kernels have fewer degrees of freedom and pre-specified tail shapes, which act as stronger regularization when the late-stage trend has become relatively clear. In addition, because only a limited number of late-stage fault detections are available to determine the long-lag behavior of the learned kernel, some sensitivity to tail extrapolation may remain. These factors may jointly explain the weaker late-stage performance of the SNN model in several cases.

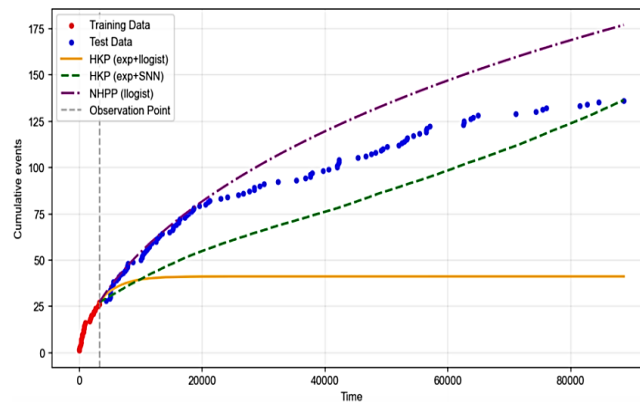


Figure 5. DS2 cumulative-fault forecasts obtained after observing the first 20% of the testing horizon.

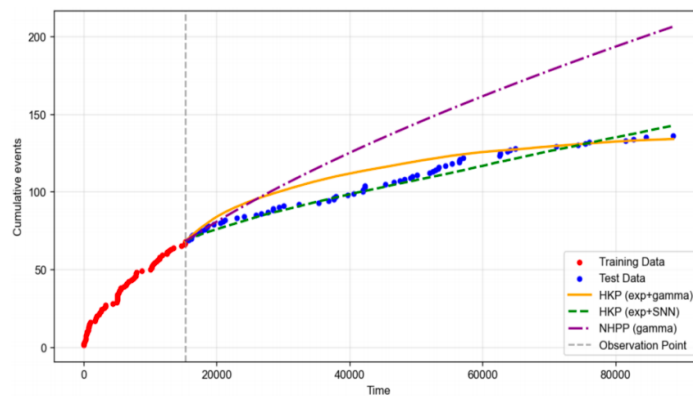


Figure 6. DS2 cumulative-fault forecasts obtained after observing the first 50% of the testing horizon.

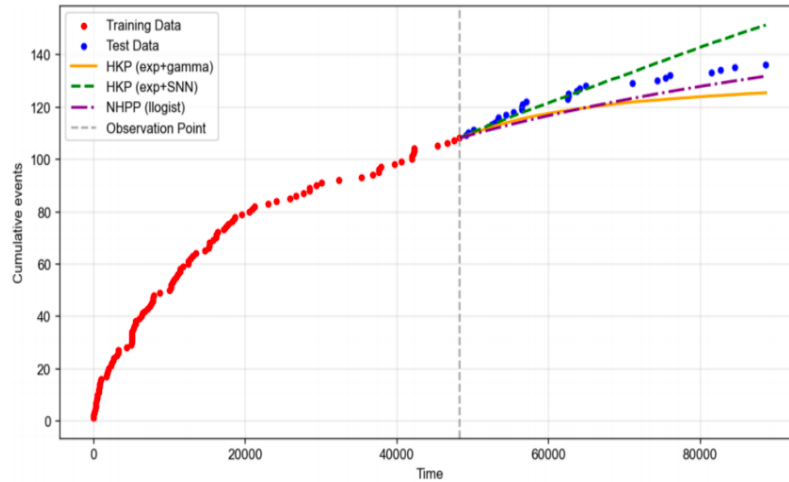


Figure 7. DS2 cumulative-fault forecasts obtained after observing the first 80% of the testing horizon.

Table 4. Predictive performances of semi-parametric and parametric HKP-based SRMs with PMSE.

Dataset	SNN	exp	gamma	pareto	lnorm	flogist	llogist	txvmax	lxvmax	txvmin	weibull
20% Observation point											
DS1	5.350	0.385	0.417	0.486	0.440	0.435	0.418	0.434	0.416	1.082	0.648
DS2	2.109	5.203	5.203	5.221	5.190	5.226	4.961	5.234	5.222	5.218	5.224
DS3	4.055	10.845	10.832	10.793	7.154	10.691	10.777	10.816	10.805	10.860	10.656
DS4	3.896	35.992	36.003	36.057	34.877	36.098	34.830	36.003	35.519	35.966	36.046
DS5	3.106	4.716	5.699	5.727	5.707	5.736	5.720	5.729	5.716	5.759	5.720
DS6	6.940	6.824	6.850	6.825	6.801	6.820	6.755	6.824	6.804	6.822	6.831
DS7	3.449	3.889	3.924	3.897	3.924	3.963	3.932	3.911	3.949	3.951	3.934
DS8	2.718	3.934	1.893	1.884	1.879	1.470	1.823	1.842	1.335	1.915	1.859
50% Observation point											
DS1	1.453	1.918	1.358	1.685	1.349	0.899	1.643	1.491	1.591	0.920	1.331
DS2	0.363	2.659	1.109	2.686	2.744	13.248	1.889	13.289	2.308	2.824	1.475
DS3	0.636	3.006	3.054	3.058	3.065	2.492	3.026	3.447	3.022	3.086	3.051
DS4	15.220	58.664	58.731	58.692	56.741	58.769	56.595	58.592	56.598	58.654	58.800
DS5	2.545	0.721	3.050	3.077	3.073	3.090	3.090	3.066	3.070	3.095	3.053
DS6	1.647	2.741	3.910	3.882	3.895	3.874	4.189	3.927	3.936	3.933	3.885
DS7	4.209	2.620	0.938	0.965	1.059	1.021	0.960	0.983	1.020	0.931	0.971
DS8	2.552	4.862	2.568	2.571	2.816	3.016	3.083	2.590	2.675	2.676	2.674
80% Observation point											
DS1	0.650	0.693	0.708	0.739	0.684	0.699	0.696	0.702	0.696	0.695	0.709
DS2	1.027	2.220	1.170	1.750	2.226	2.115	2.244	2.119	2.226	2.131	1.149
DS3	3.205	1.040	1.362	1.388	1.364	1.424	1.348	1.397	0.813	1.382	1.448
DS4	3.840	2.177	3.724	3.994	3.756	3.764	3.815	4.088	3.754	4.018	4.032
DS5	1.483	0.718	1.735	1.659	1.700	1.702	1.711	1.708	1.680	1.692	1.708
DS6	2.199	0.688	1.330	1.302	1.349	1.324	1.515	1.306	1.378	1.308	1.359
DS7	3.913	3.399	2.531	2.508	2.463	2.459	2.537	2.542	2.508	2.464	2.451
DS8	1.651	1.042	1.891	1.908	1.898	1.878	1.901	1.929	1.848	1.896	1.944

From an overall viewpoint, our proposed semi-parametric SRM is not always the best SRM for every dataset and every observation point. However, it outperforms most of the parametric HKP-based SRMs in pairwise PMSE comparisons. This suggests that the SNN excitation kernel can reduce the dependence on manually selecting a specific parametric kernel form. In practical applications, the true excitation pattern of software fault detections is usually unknown in advance. Therefore, a data-adaptive SNN kernel provides

a useful and flexible alternative to fixed parametric kernels.

Table 5 compares our proposed semi-parametric HKP-based SRM with representative NHPP-based SRMs in terms of PMSE. From these results, the semi-parametric SRM outperforms the NHPP-based SRMs in most experimental settings. In particular, at the 50% observation point, our proposed SRM obtains smaller PMSE values on seven out of eight datasets. This result indicates that when a moderate amount of fault-detection data is available, our proposed SRM can effectively use historical fault information and learn a more suitable excitation pattern than deterministic NHPP-based SRMs.

Table 5. Predictive performances of semi-parametric HKP-based SRMs and NHPP-based SRMs with PMSE.

Dataset	Semi-parametric HKP-based SRM	NHPP-based SRMs	
	PMSE	Best baseline model	PMSE
20% Observation point			
DS1	5.350	exp	0.413
DS2	2.109	llogist	1.740
DS3	4.055	exp	10.968
DS4	3.896	txvmax	9.379
DS5	3.106	tlogist	7.198
DS6	6.940	gamma	6.917
DS7	3.449	llogist	2.268
DS8	2.718	gamma	11.694
50% Observation point			
DS1	1.453	exp	1.766
DS2	0.363	gamma	3.889
DS3	0.636	exp	3.051
DS4	15.220	lnorm	45.212
DS5	2.545	tlogist	3.857
DS6	1.647	txvmin	4.944
DS7	4.209	tlogist	2.156
DS8	2.552	tlogist	6.061
80% Observation point			
DS1	0.650	exp	0.700
DS2	1.027	llogist	0.977
DS3	3.205	lxvmax	0.554
DS4	3.840	llogist	3.212
DS5	1.483	tlogist	1.320
DS6	2.199	tlogist	4.545
DS7	3.913	txvmax	1.072
DS8	1.651	lxvmax	2.165

Overall, the comparison demonstrates that our proposed semi-parametric SRM is not only a flexible alternative to parametric HKP-based SRMs, but also a competitive model against traditional NHPP-based SRMs. The results suggest that incorporating self-excitation through an SNN kernel is practically valuable for software reliability prediction.

6. Conclusion

This paper proposed a semi-parametric HKP-based software reliability model by combining an exponential time-varying base intensity with an SNN excitation kernel. Our proposed SRM was designed to capture two important characteristics of software fault-detection processes: the long-term decreasing trend of the background fault-detection rate and the short-term history-dependent self-exciting effect caused by previous fault detections.

Compared with conventional parametric HKP-based SRMs, our proposed semi-parametric SRM avoids relying on a fixed pre-specified excitation kernel. This is important because the most suitable parametric kernel may vary across datasets and software projects. By using an SNN kernel, our proposed SRM can adaptively represent different excitation shapes while preserving the explicit kernel-based structure of the Hawkes process. In addition, because the ReLU-based SNN kernel is piecewise exponential, its integrated form can be evaluated efficiently, which makes likelihood-based training computationally tractable.

The experimental results on eight software fault-detection datasets demonstrated the practical effectiveness of our proposed SRM. In comparison with ten parametric HKP-based SRMs, our proposed semi-parametric SRM achieved strong predictive performance, particularly at the 20% and 50% observation points. This suggests that the flexibility of the SNN excitation kernel is especially useful in early- and middle-stage prediction, where the available fault-detection data are still limited and the underlying excitation pattern is not sufficiently clear. In the late-stage setting, some parametric kernels and NHPP-based SRMs can also provide accurate predictions because the remaining trend becomes easier to capture. The comparison with representative NHPP-based SRMs further showed that our proposed semi-parametric SRM is competitive against traditional Markovian models. In particular, at the 50% observation point, our proposed SRM outperformed NHPP-based SRMs on seven out of eight datasets. This result indicates that incorporating history-dependent self-excitation through an SNN kernel can provide practical prediction benefits when a moderate amount of fault-detection data is available.

Overall, our proposed semi-parametric HKP-based SRM provides a flexible and interpretable modeling framework for software reliability prediction. It can serve as an effective alternative to fixed parametric HKP-based SRMs and a competitive extension of traditional NHPP-based SRMs.

Future work will extend the present framework to other parametric base-intensity forms, including Weibull- and Pareto-type functions, in order to represent different decay and tail behaviors. We will also investigate non-parametric base intensities, such as spline-based or other flexible forms. A further direction is to construct fully non-parametric HKP-based SRMs in which both the base intensity and the excitation kernel are estimated flexibly, while preserving positivity, interpretability, and computational tractability. These extensions will help clarify how the choice of the base intensity affects prediction across different software testing environments.

Conflicts of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

The first author, Nanxiang Qiu, was partially supported by JST SPRING, Grant Number JPMJSP2132.

AI Disclosure

The authors confirm that no Artificial Intelligence (AI) generated text is used throughout this manuscript.

References

- Abdel-Ghaly, A.A., Chan, P.Y., & Littlewood, B. (1986). Evaluation of competing software reliability predictions. *IEEE Transactions on Software Engineering*, SE-12(9), 950-967. <https://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. *Frontiers in Reliability*, 4, 1-18. https://doi.org/10.1142/9789812816580_0001

- Chen, Y., & Singpurwalla, N.D. (1997). Unification of software reliability models by self-exciting point processes. *Advances in Applied Probability*, 29(2), 337-352. <https://doi.org/10.2307/1428006>
- Cui, L., Hawkes, A., & Yi, H. (2020). An elementary derivation of moments of Hawkes processes. *Advances in Applied Probability*, 52(1), 102-137. <https://doi.org/10.1017/apr.2019.53>
- Erto, P., Giorgio, M., & Lepore, A. (2020). The generalized inflection S-shaped software reliability growth model. *IEEE Transactions on Reliability*, 69(1), 228-244. <https://doi.org/10.1109/TR.2018.2869466>
- Goel, A.L. (1985). Software reliability models: assumptions, limitations and applicability. *IEEE Transactions on Software Engineering*, SE-11(12), 1411-1423. <https://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 Transactions on Reliability*, R-28(3), 206-211. <https://doi.org/10.1109/TR.1979.5220566>
- Gokhale, S.S., & Trivedi, K.S. (1998). Log-logistic software reliability growth model. In *Proceedings of the Third IEEE International High-Assurance Systems Engineering Symposium* (pp. 34-41). IEEE. Computer Society, Washington, DC, USA. <https://doi.org/10.1109/HASE.1998.731593>
- Hawkes, A.G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1), 83-90. <https://doi.org/10.1093/biomet/58.1.83>
- Hawkes, A.G., & Oakes, D. (1974). A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11(3), 493-503. <https://doi.org/10.2307/3212693>
- Iwamoto, K., Dohi, T., & Okamura, H. (2022). Software reliability assessment with a self-exciting Hawkes process. In *Proceedings of the 10th International Symposium on Advanced Reliability and Maintenance Modeling (APARM 2022)*. <https://aparm2022.conf.tw/site/userdata/1420/Conference%20Program%20Book.pdf>
- Joseph, S., Kashyap, L.D., & Jain, S. (2022). Shallow neural Hawkes: non-parametric kernel estimation for Hawkes processes. *Journal of Computational Science*, 63, 101754. <https://doi.org/10.1016/j.jocs.2022.101754>
- Lallouache, M., & Challet, D. (2016). The limits of statistical significance of Hawkes processes fitted to financial data. *Quantitative Finance*, 16(1), 1-11. <https://doi.org/10.1080/14697688.2015.1068442>
- Lindström, A., Lindgren, S., & Sainudiin, R. (2023). Hawkes processes on social and mass media: a causal study of the BlackLivesMatter movement in the summer of 2020. In *Proceedings of the 12th International Conference on Data Science, Technology and Applications* (pp. 77-88). SciTePress, Rome, Italy. <https://doi.org/10.5220/0012089500003541>
- Littlewood, B. (1984). Rationale for a modified Duane model. *IEEE Transactions on Reliability*, R-33(2), 157-159. <https://doi.org/10.1109/TR.1984.5221762>
- Lyu, M.R. (1996). *Handbook of software reliability engineering*. IEEE Computer Society Press, McGraw-Hill.
- Mei, H., & Eisner, J.M. (2017). The neural Hawkes process: a neurally self-modulating multivariate point process. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds) *Advances in Neural Information Processing Systems* (Vol. 30, pp. 6754-6764). Curran Associates, Inc.
- Musa, J.D. (1979). *Software reliability data* [Technical report]. Rome Air Development Center. : <https://github.com/Derek-Jones/Reliability-data>
- Musa, J.D., & Okumoto, K. (1984). A logarithmic Poisson execution time model for software reliability measurement. In *Proceedings of the 7th International Conference on Software Engineering* (pp. 230-238). Orlando.
- Musa, J.D., Iannino, A., & Okumoto, K. (1987). *Software reliability: measurement, prediction, application*. McGraw-Hill, United States.

- Ohba, M. (1984). Inflection S-shaped software reliability growth model. In: Osaki, S., Hatoyama, Y. (eds) *Stochastic Models in Reliability Theory* (pp. 144-162). Springer, Berlin, Heidelberg. https://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. *Journal of Systems and Software*, 82(3), 535-543. <https://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* (pp. 100-107). IEEE. Pasadena, CA, USA. <https://doi.org/10.1109/ISSRE.2013.6698909>
- Okamura, H., Dohi, T., & Osaki, S. (2013). Software reliability growth models with normal failure time distributions. *Reliability Engineering & System Safety*, 116, 135-141. <https://doi.org/10.1016/j.ress.2012.02.002>
- Qiu, N., Dohi, T., & Okamura, H. (2024). Evaluation of software reliability models based on Hawkes processes. In *Recent Advances in Reliability and Maintenance Modeling* (pp. 192-200). CRC Press. <https://doi.org/10.1201/9781003491309-23>
- Qiu, N., Dohi, T., Zheng, J., & Okamura, H. (2025). Self-exciting software reliability models with Pareto base intensity function and their applications. In *2025 25th International Conference on Software Quality, Reliability and Security* (pp. 416-427). IEEE. Hangzhou, China. <https://doi.org/10.1109/QRS65678.2025.00049>
- Reinhart, A., & Greenhouse, J. (2018). Self-exciting point processes with spatial covariates: modelling the dynamics of crime. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(5), 1305-1329. <https://doi.org/10.1111/rssc.12277>
- Rizoiu, M.-A., Soen, A., Li, S., Calderon, P., Dong, L., Menon, A.K., & Xie, L. (2022). Interval-censored Hawkes processes. *Journal of Machine Learning Research*, 23(338), 1-84. <https://jmlr.org/papers/v23/21-0917.html>
- Wang, Q. (2015). *Applications of Hawkes' process in finance* [Master's thesis, Tilburg University].
- Xie, M. (1991). *Software reliability modeling*. World Scientific.
- Zhang, L.N., Liu, J.W., & Zuo, X. (2020). Survival analysis of failures based on Hawkes process with Weibull base intensity. *Engineering Applications of Artificial Intelligence*, 93, 103709. <https://doi.org/10.1016/j.engappai.2020.103709>
- Zhao, M., & Xie, M. (1996). On maximum likelihood estimation for a general nonhomogeneous Poisson process. *Scandinavian Journal of Statistics*, 23(4), 597-607. <https://www.jstor.org/stable/4616427>