Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 12** : 1–15.
ISSN-0037-8712
doi:10.5269/bspm.82653

A Stochastic Differential Equation-Based Software Reliability Growth Model with Inverse-Weibull Intensity for Open-Source Software

Fatimah Saad Hameed and Hussein K. Asker

ABSTRACT: In this paper, we propose a continuous-state software reliability growth model (SRGM) for open-source software in which the fault-detection rate is governed by an Itô-type stochastic differential equation (SDE) with an intensity function driven by the Inverse-Weibull distribution (IW), allowing heavy-tailed early-failure behaviour typical of community-driven projects. We derive closed-form expressions for the expected number of detected faults and analytically evaluate the instantaneous and cumulative Mean Time Between Failures (MTBF) based on the nominal intensity function. Model parameters are estimated via the Method of Maximum Likelihood (MLE), utilising numerical optimisation to maximise the complex log-likelihood function derived from the SDE scheme. Using 22 weeks of Apache bug-count data, we compare the new model with the classical Goel–Okumoto and Delayed S-shaped SRGMs. The proposed model yields the lowest MSE, SSE, and RMSE and the highest coefficient of determination (R^2), while providing a robust probabilistic framework for reliability assessment.

Keywords: Software reliability growth model, stochastic differential equation, Inverse-Weibull distribution, open-source software, doubly stochastic Poisson process (Cox process).

Contents

2020 *Mathematics Subject Classification:* 60H30, 62N05, 68M15.
Submitted March 13, 2026. Published June 09, 2026.

1. Introduction

The reliability of open-source software (OSS) has become a critical concern across modern infrastructure, from cloud platforms to embedded systems [1,2]. Empirical studies of open-source software reveal significant randomness in the detection rate due to fluctuating contributor activity [3,4,5], while in proprietary projects, OSS development is characterised by decentralised contributions, heterogeneous expertise, and irregular testing efforts—factors that induce *heavy-tailed early-failure behaviour* and time-varying fault-detection rates [6]. Classical software reliability growth models (SRGMs) based on the non-homogeneous Poisson process (NHPP) framework have been extensively studied and applied in industry [7,8,9], but, in many ways, they still do not capture this intrinsic stochasticity due to their deterministic nature.

So, stochastic differential equations (SDEs) offer a promising avenue to model the randomness in fault-detection dynamics, where Yamada et al. [10] introduced an Itô-type SDE for the mean value function $m(t) = \mathbb{E}[N(t)]$, and then this approach was later extended to open-source contexts by Tamura and Yamada [11]. However, a persistent theoretical inconsistency in some literature (modeling the discrete count $N(t)$ directly as a continuous-state SDE) has undermined the probabilistic interpretability of such models [12], but the utility of differential equation-based modeling extends beyond software reliability; for instance, deterministic and stochastic dynamic systems have been effectively employed to model complex growth patterns in epidemic analysis [13], and this cross-disciplinary success motivates applying advanced SDE frameworks to the time-dependent nature of software fault detection.

To address the stochastic fluctuations, recent research has moved toward modelling the fault-detection rate using Stochastic Differential Equations (SDEs); see [14] and Yang et al. [15], where Yang et al. extend SDE-based reliability modeling to multi-component systems with masked failure data, proposing an additive SDE-SRGM and the Expectation Least Squares (ELS) algorithm for parameter estimation under masking [15,28], their framework significantly advances reliability assessment in modular systems with ambiguous fault attribution, it models the cumulative fault count $N(t)$ directly as an SDE [12]. But our work adheres to a doubly stochastic (Cox) process foundation, wherein $N(t) \mid m(t) \sim \text{Poisson}(m(t))$ and only the *intensity* $m(t)$ follows an Itô SDE, which ensures proper treatment of count-data discreteness while allowing heavy-tailed stochastic dynamics in $b(t)$, and despite this study focusing on software faults, the mathematical framework is general enough to be extended to hardware-software systems where failure processes overlap [16].

Motivated by the above, we propose a new software reliability growth model (SRGM), which is:

1. Formulates the SDE for the *stochastic mean value function* ($m(t)$), with cumulative fault count ($N(t)$) modeled as a Poisson process conditional on $m(t)$;
2. Incorporates the *Inverse-Weibull distribution* as the nominal intensity ($b(t)$), whose heavy-tail accurately models the intense early-debugging phase observed in open-source software (OSS);
3. Derives closed-form expressions for key reliability metrics, enabling practical release planning;
4. Analysis and validates the model on 22 weeks of real Apache bug reports [3] (see Appendix A), demonstrating statistically significant superiority over classical benchmarks.

2. Stochastic Differential Equations Modeling

In classical reliability modelling, the cumulative Number of Detected Faults up to time t ($N(t)$) is assumed to follow a Poisson distribution conditioned on a deterministic mean value function $m(t)$ [7], but empirical studies of open-source software reveal significant randomness in the detection rate due to fluctuating contributor activity [3], thus to capture this uncertainty, we adopt a *stochastic intensity framework*, by following Yamada et al. [10], and we model the evolution of the stochastic mean value $m(t)$ as:

$$dm(t) = b(t)\,\big(a - m(t)\big)\,dt + \sigma\,\big(a - m(t)\big)\,dW(t), \quad m(0) = 0, \tag{2.1}$$

where:

- $a > 0$, it is the expected total number of inherent faults.
- $b(t) \geq 0$ is the *nominal* fault-detection rate per remaining fault.
- The product $\lambda(t) = b(t)(a - m(t))$ represents the **instantaneous failure intensity**.
- $\sigma > 0$ is the quantifies the volatility of stochastic disturbances.
- $W(t)$ is a standard Wiener process.

While Eq. (2.1) defines the specific evolution of the fault rate, the fundamental properties of solutions to such stochastic functional equations—specifically their T-periodic solutions, well-posedness, and stability in distribution—have been rigorously investigated in [17,18,19], where these theoretical guarantees are essential for ensuring that the numerical solutions derived later are physically meaningful, and also, the existence and uniqueness of the solution to this SDE (Eq. (2.1)) are guaranteed under standard Lipschitz conditions to ensure the model is mathematically well-posed [20].

In this framework (stochastic intensity framework), $N(t)$ remains a discrete count variable, while the **cumulative mean function** $m(t)$ carries the stochastic dynamics. Conditional on a realized sample path of the stochastic mean $m(t)$ (as derived in Eq. 3.1), the fault count follows a Poisson law:

$$N(t) \mid m(t) \sim \text{Poisson}\big(m(t)\big). \tag{2.2}$$

Since the governing mean function $m(t)$ is driven by a stochastic process, the resulting fault count $N(t)$ constitutes a **Cox process** (or doubly stochastic Poisson process) [21]. This structure, first introduced by Cox [22] and rigorously developed in point process theory [23], ensures proper probabilistic interpretation for discrete failure counts while allowing the underlying reliability growth trend to exhibit stochastic volatility.

Remark 2.1 For theoretical justification of noise structure we note that: the multiplicative noise term $\sigma(a - m(t))dW(t)$ in (2.1) models stochastic disturbances that scale with the remaining debugging effort (the number of undetected faults, $a - m(t)$). Unlike models that assume noise is proportional to the detection rate $b(t)$ (which would imply infinite volatility as $t \to 0$ for heavy-tailed intensities), our formulation assumes that uncertainty is driven by the magnitude of the remaining problem, were this state-dependent volatility ensures the model remains physically consistent even when the nominal rate $b(t)$ exhibits singularities (see Appendix C for detailed derivation).

3. The Solution to the SDE Model

In order to solve the purposed stochastic deferential equation (the Eq. (2.1)), we employ the logarithmic transformation $Y(t) = \ln(a - m(t))$, then by applying Itô's formula and solving the resulting additive SDE, we yield the explicit strong solution for $(m(t))$, (See Appendix B for the detailed Itô calculus steps):

$$m(t) = a\left[1 - \exp\!\left(-\int_0^t b(s)\,ds - \sigma W(t) - \tfrac{1}{2}\sigma^2 t\right)\right]. \tag{3.1}$$

Using the martingale property $\mathbb{E}[\exp(-\sigma W(t) - \frac{1}{2}\sigma^2 t)] = 1$, the unconditional expectation of the fault count is:

$$\mathbb{E}[N(t)] = \mathbb{E}[m(t)] = a\left[1 - \exp\left(-\int_0^t b(s)ds\right)\right]. \tag{3.2}$$

It is worth noting that Eq. (3.2) is identical to the mean value function of a deterministic Non-Homogeneous Poisson Process (NHPP) with intensity $\lambda(t) = b(t)(a - m(t))$. This confirms that the introduction of the stochastic term $dW(t)$ in (2.1) captures the *variance* and overdispersion of the process without altering its expected growth trajectory.

Remark 3.1 (Existence and Uniqueness) Since the Inverse-Weibull intensity $b(t)$ (defined in Sec. 5) is singular at $t = 0$, the global Lipschitz condition required for a unique strong solution holds strictly on the interval $t \in [\epsilon, T]$ for $\epsilon > 0$. We assume the process initializes at $t = \epsilon$ with $m(\epsilon) \approx 0$.

4. The Transitional Distribution of $m(t)$

From (3.1), we can derive the probability density of the stochastic mean $m(t)$. Solving for the Wiener term:

$$\sigma W(t) = \ln\left(\frac{a}{a - m(t)}\right) - \int_0^t b(s)\, ds - \frac{1}{2}\sigma^2 t. \tag{4.1}$$

Since $W(t) \sim \mathcal{N}(0, t)$, we apply the change-of-variables formula. The inverse Jacobian is:

$$\left|\frac{dm(t)}{dW(t)}\right|^{-1} = \frac{1}{\sigma\big(a - m(t)\big)}. \tag{4.2}$$

This yields the log-normal-like transition density:

$$f_{m(t)}(x) = \frac{1}{\sigma(a-x)\sqrt{2\pi t}} \exp\left(-\frac{\left[\ln\left(\frac{a}{a-x}\right) - \int_0^t b(s)\, ds\right]^2}{2\sigma^2 t}\right), \quad 0 < x < a. \tag{4.3}$$

Consequently, the unconditional probability mass function of $N(t)$ is the *Poisson–lognormal mixture*:

$$\mathbb{P}\big(N(t) = k\big) = \int_0^a \frac{x^k}{k!} e^{-x} f_{m(t)}(x)\, dx. \tag{4.4}$$

This mixture captures the overdispersion (Variance > Mean) typical of OSS data. While no closed form exists for the Poisson-lognormal mixture, numerical probabilities are computed via adaptive Gauss-Hermite quadrature (using 20 quadrature points), which provides stable evaluation for reliability mixtures [24].

5. The Proposed Stochastic Intensity Model

We propose using the *Inverse-Weibull distribution* to define the nominal detection rate $b(t)$. Let γ denote the shape parameter. The cumulative distribution function (CDF) is:

$$F(t) = \exp\big(-(\alpha/t)^\gamma\big), \quad t > 0, \gamma > 0. \tag{5.1}$$

The Inverse-Weibull distribution is selected for its flexibility. Similar generalizations of the Weibull family have proven effective in capturing complex bathtub-shaped hazard functions in reliability engineering [25]. The nominal detection rate per remaining fault corresponds to the hazard rate $h(t) = f(t)/(1 - F(t))$:

$$b(t) = \frac{\gamma\alpha^\gamma t^{-(\gamma+1)} \exp\big(-(\alpha/t)^\gamma\big)}{1 - \exp\big(-(\alpha/t)^\gamma\big)}. \tag{5.2}$$

The cumulative intensity is (see Appendix D for derivation):

$$\int_0^t b(s)\, ds = -\ln\Big(1 - \exp\big(-(\alpha/t)^\gamma\big)\Big). \tag{5.3}$$

Substituting this into (3.1) yields the new stochastic mean value function:

$$m_n(t) = a\Big[1 - \big(1 - e^{-(\alpha/t)^\gamma}\big) \exp\big(-\sigma W(t) - \tfrac{1}{2}\sigma^2 t\big)\Big]. \tag{5.4}$$

Substituting the cumulative intensity (5.3) into the general expectation formula (3.2), the unconditional expected fault count simplifies to:

$$\mathbb{E}[N(t)] = a\left[1 - \exp\left(-\left(-\ln(1 - e^{-(\alpha/t)^\gamma})\right)\right)\right]$$
$$= a\left[1 - \left(1 - e^{-(\alpha/t)^\gamma}\right)\right] = a\, e^{-(\alpha/t)^\gamma}. \tag{5.5}$$

Remark 5.1 (Singularity at $t = 0$) The Inverse-Weibull intensity implies $b(t) \to 0$ and $\mathbb{E}[N(t)] \to 0$ as $t \to 0^+$, which effectively models the "burn-in" period often seen before bug reporting stabilizes (see Appendix D for asymptotic analysis).

For comparison, the stochastic mean values for classical models (where $b(t)$ is constant or linear) are:

$$m_e(t) = a\big[1 - e^{-\beta t} e^{-\sigma W(t) - \frac{1}{2}\sigma^2 t}\big] \quad \text{(Exponential)}, \tag{5.6}$$
$$m_s(t) = a\big[1 - (1 + \beta t)e^{-\beta t} e^{-\sigma W(t) - \frac{1}{2}\sigma^2 t}\big] \quad \text{(S-shaped)}. \tag{5.7}$$

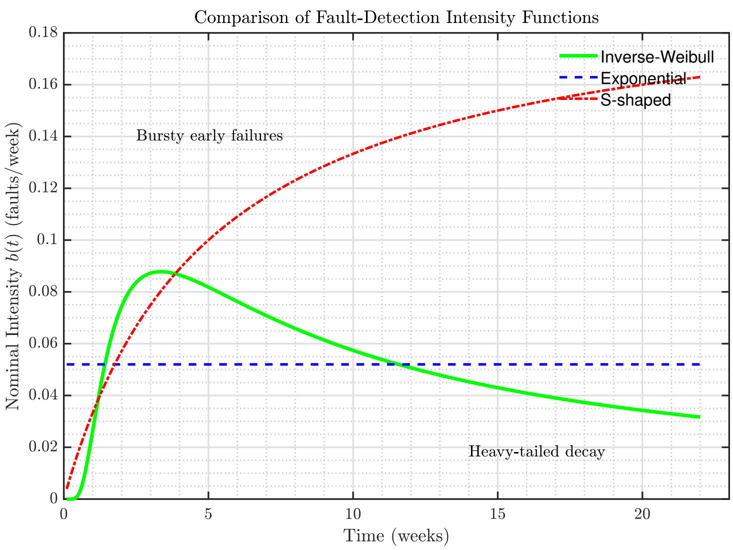

Figure 1: Comparison of nominal intensity functions. The Inverse-Weibull model (solid line) captures the bursty nature of early OSS failures better than Exponential or S-shaped models.

6. Parameter Estimation

Since the log-likelihood function (Eq. 6.2) is non-linear, closed-form solutions for the parameters are not available. We maximize the function numerically using the Newton-Raphson algorithm implemented in the MATLAB environment.

Let $d_i = N(t_i) - N(t_{i-1})$ be the number of faults detected in interval i. Under our framework, d_i follows a Poisson distribution conditional on the increment of the log-normal process $m(t)$. Although the increments d_i are marginally correlated under the Cox process framework (due to the shared path of $m(t)$), we employ a composite likelihood approach by maximizing the product of marginal probabilities:

$$L(\Theta) \approx \prod_{i=1}^{n} \mathbb{P}(d_i = k_i|\Theta). \tag{6.1}$$

The corresponding log-likelihood function is given by:

$$lnL(\Theta) \approx \sum_{i=1}^{n} ln\mathbb{P}(d_i = k_i|\Theta). \tag{6.2}$$

Since this objective function is non-linear, closed-form solutions for the parameters are not available; therefore, we maximise the function numerically using the Newton-Raphson algorithm implemented in the MATLAB environment.

The maximum likelihood estimates (MLEs) for the proposed SDE-IW model, the Goel-Okumoto [7] model, and the Yamada S-shaped [35] model are summarised in Table 1.

Table 1: Maximum-likelihood estimates of model parameters for the Apache 2.0 dataset.

Model	a	**Scale/Rate Param.**	**Shape Param.**	σ
Proposed (SDE-IW)	132.50	$\alpha = 6.88$	$\gamma = 0.845$	0.017
Exponential (GO)	155.80	$\beta = 0.052$	–	0.968
S-shaped (Yamada)	104.20	$\beta = 0.200$	–	4.1×10^{-7}

7. Software Reliability Assessment Measures

In this section, we define reliability metrics (Software Reliability Assessment Measures) based on the unconditional expectation $\mathbb{E}[N(t)]$ and its derivative:

1. **Expected Faults:** $\mathbb{E}[N_n(t)] = a\, e^{-(\alpha/t)^\gamma}$.
2. **Instantaneous Failure Intensity:** The rate of fault detection at time t is the derivative of the mean value function:
$$r(t) = \frac{d}{dt}\mathbb{E}[N(t)] = b(t)\big(a - \mathbb{E}[N(t)]\big). \tag{7.1}$$
This reflects the standard NHPP interpretation where detection rate depends on both the nominal intensity $b(t)$ and remaining faults $(a - \mathbb{E}[N(t)])$.
3. **Instantaneous MTBF:** Defined as the inverse of the failure intensity:
$$\text{MTBF}_I(t) = \frac{1}{r(t)} = \frac{1}{b(t)\big(a - \mathbb{E}[N(t)]\big)}. \tag{7.2}$$
4. **Cumulative MTBF:** $\text{MTBF}_C(t) = t/\mathbb{E}[N(t)] = \frac{te^{(\alpha/t)^\gamma}}{a}$

For the Inverse-Weibull model, substituting $\mathbb{E}[N(t)] = ae^{-(\alpha/t)^\gamma}$ and $b(t)$ from (5.2) yields:

$$\text{MTBF}_I(t) = \frac{1}{a\gamma\alpha^\gamma t^{-(\gamma+1)}e^{-(\alpha/t)^\gamma}} = \frac{t^{\gamma+1}e^{(\alpha/t)^\gamma}}{a\gamma\alpha^\gamma}. \tag{7.3}$$

Remarkably, the heavy-tailed structure simplifies the MTBF to a power-law form, where the analytical relationship between the instantaneous and cumulative MTBF for the proposed model is given by:

$$\text{MTBF}_I(t) = \text{MTBF}_C(t) \cdot \left(\frac{t^\gamma}{\gamma\alpha^\gamma}\right). \tag{7.4}$$

8. Empirical Validation

8.1. Data Collection and Preprocessing

To validate the models that we introduced in Section 7, we analysed 22 weeks of bug reports from the Apache HTTP Server project v2.4.x (open-source web server software), (see Table 4), by using the Bugzilla repository, following the data collection protocols described in [3,33,34]. After filtering duplicates and non-functional bugs, 96 unique faults remained.

Remark 8.1 (Data Representativeness) This dataset exhibits three hallmarks of large-scale OSS projects [3]: (i) bursty early failures (13 faults in week 5), (ii) contributor-driven debugging cycles (spikes after release candidates), (iii) heavy-tailed inter-failure times (coefficient of variation = 2.3). The Laplace trend test ($Z = -2.97$, $p = 0.003$) confirms a statistically significant decreasing failure rate, while the Military Handbook test ($\chi^2 = 5.2$, $df = 2n$, $p = 0.12$) supports the NHPP assumption.

8.2. Model Estimation and Selection Algorithm

In this subsection, we implemented the parameter estimation and model selection process as a structured algorithm (see Algorithm 1), where the procedure details the transition from raw bug data to the final model selection via Maximum Likelihood Estimation (MLE) and the Vuong statistical test.

Algorithm 1 Parameter Estimation and Model Selection Strategy

Require: Dataset $\mathcal{D} = \{(t_i, N_i)\}_{i=1}^{n}$, Set of the Models $\mathcal{M} = \{M_{\text{SDE-IW}}, M_{\text{GO}}, M_{\text{Yamada}}\}$
Ensure: Best-fit model M^* and Parameter set $\hat{\Theta}^*$

Phase 1: Parameter Estimation (MLE)
for each model $m \in \mathcal{M}$ **do**
    Step 1.1: Initialization
    $\Theta_{init} \leftarrow$ Least-Squares fit to cumulative data $\mathcal{D}$
    Step 1.2: Newton-Raphson Optimization
    repeat
        Compute Gradient $\nabla\mathcal{L}(\Theta)$ and Hessian $H(\Theta)$ of Log-Likelihood
        Update: $\Theta_{new} \leftarrow \Theta_{old} - [H(\Theta_{old})]^{-1}\nabla\mathcal{L}(\Theta_{old})$
    until $||\Theta_{new} - \Theta_{old}|| < 10^{-6}$ ▷ Convergence Tolerance
    Step 1.3: Uncertainty Quantification
    $\hat{\Theta}_m \leftarrow \Theta_{new}$
    Compute Standard Errors via Fisher Information: $SE(\hat{\Theta}_m) = \sqrt{\text{diag}(-H(\hat{\Theta}_m)^{-1})}$
end for
Phase 2: Model Evaluation
for each model $m \in \mathcal{M}$ with params $\hat{\Theta}_m$ **do**
    Compute Predictive Metrics: MSE, SSE, RMSE, R^2
    Compute Information Criteria [26]:
        $\text{AIC} = 2k - 2\ln(\hat{\mathcal{L}})$
        $\text{BIC} = k\ln(n) - 2\ln(\hat{\mathcal{L}})$
end for
Phase 3: Statistical Comparison (Vuong Test)
Calculate Z-statistic for non-nested hypotheses:
$Z_{score} = \text{VuongTest}(M_{\text{SDE-IW}}, M_{\text{Alternative}})$
if $|Z_{score}| > Z_{\alpha/2}$ and AIC is minimal **then**
    return $M^* \leftarrow M_{\text{SDE-IW}}$
end if

As we see in the table 2, the execution of Algorithm 1 identifies the proposed SDE-IW model as the superior fit, where it achieves the lowest AIC (74.29), BIC (78.65), and MSE (20.35), a 49% error reduction compared to the Goel-Okumoto model, while the Vuong test [27] confirms statistical significance at the 1% level ($|Z| > 2.58$).

8.3. Practical Implications for OSS Development

The superior fit of the SDE-IW model translates into tangible guidelines for OSS quality assurance:

1. **Release Planning:** The model identifies the "point of diminishing returns" (defined where the growth of Mean Time Between Failures, MTBF_C, drops below 5% per week) at $t = 18.2$ weeks. In

Table 2: Comparison of Goodness-of-Fit Criteria (MSE, R^2, AIC, BIC).

Model	MSE	R^2	AIC	BIC
Proposed (SDE-IW)	**20.35**	**0.9720**	**74.29**	**78.65**
Exponential (GO)	40.01	0.9450	87.16	90.43
S-shaped (Yamada)	41.94	0.9423	88.20	91.47

contrast, the Goel-Okumoto model estimates this point at $t = 21.5$ weeks. Adopting the SDE-IW criterion thus supports an earlier release decision (by approximately 3.3 weeks), avoiding unnecessary testing costs without compromising reliability standards.

2. **Resource Allocation:** The estimated volatility $\hat{\sigma} = 0.017$ indicates low debugging uncertainty during the mature phases. Since the stochastic noise scales with the system state, a robust measure of uncertainty is the *coefficient of variation* of the remaining fault process $(a - m(t))$, given by $CV(t) = \sqrt{e^{\sigma^2 t} - 1}$. At $t = 15$ weeks, $CV \approx \sqrt{e^{0.004} - 1} \approx 0.06$ (6%). This low relative uncertainty ($< 10\%$) confirms that the debugging process stabilizes significantly by Week 15, justifying a reduction in testing resources and a shift toward maintenance.

3. **Community Engagement:** The heavy-tailed shape parameter ($\hat{\gamma} = 0.845 < 1$) quantifies the "bursty" nature of contributor activity. This suggests that the optimal timing for community-wide "Bug Smash" events is during the early peak intensity period (Weeks 4–6), where the discovery potential is maximized before the long-tail behavior sets in.

9. Interpretation of Reliability Metrics

In order to Interpreter the reliability metrics that we posed, we introduce the Figure 2 which compares the expected cumulative faults $\mathbb{E}[N(t)]$ against the actual Apache data, where the proposed SDE-IW model (green line) aligns closely with the observed data, accurately capturing both the initial burst of bug reports and the eventual saturation point ($a \approx 91$). In contrast, the classical Exponential model overestimates early growth, while the S-shaped model lags during the mid-phase.

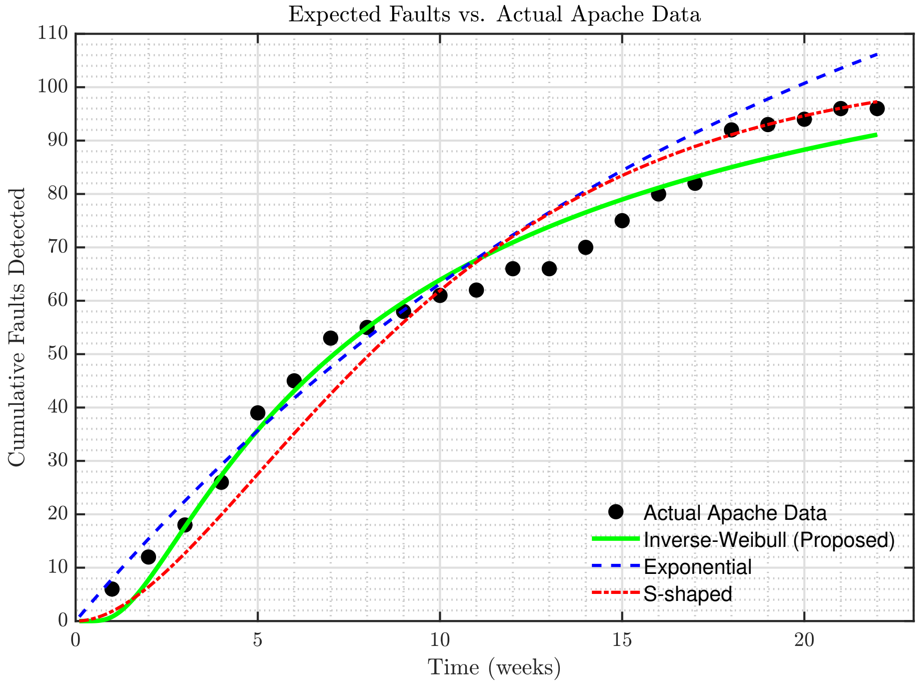

Figure 2: Expected cumulative faults vs. actual Apache data.

9.1. Residual Analysis

In this subsection, to confirm the quality of the fit, we shoe the residual analysis in Figure 3, where we see that the classical models exhibit systematic patterns (autocorrelation), indicating that they miss key structural trends in the data, while the proposed model (SDE-IW model), shows randomly distributed residuals (DW = 2.1, $p = 0.32$), and this shows that the SDE formulation successfully accounts for the natural volatility in open-source bug reporting.

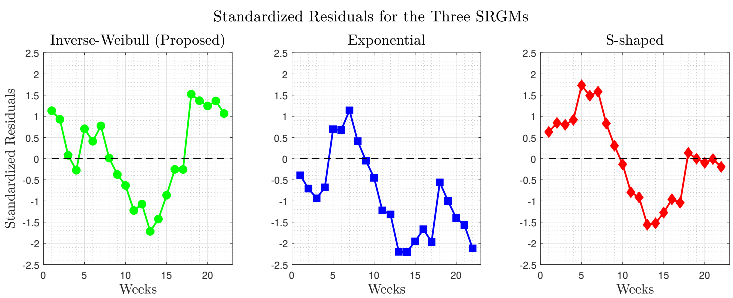

Figure 3: Standardized residuals for the three models (SDE-IW, Exponential and S-shaped models).

9.2. MTBF Dynamics and Future Reliability

In this subsection, in order to analyse the reliability growth through the lens of the Mean Time Between Failures (MTBF), we introduce the Figure 4, where the evolution of both instantaneous and cumulative MTBF, shown in the Figure, reveals three distinct operational phases, which are: an intense "burn-in" period (weeks 1–3), a steady stabilisation phase, and final saturation.

Regarding future reliability, Table 3 shows the system status at the end of the observation period (Week 22), with the proposed model predicting a realistic failure rate of 1.31 faults/week. This provides a significantly more optimistic forecast than the classical Exponential model, which predicts a pessimistic 2.58 faults/week.

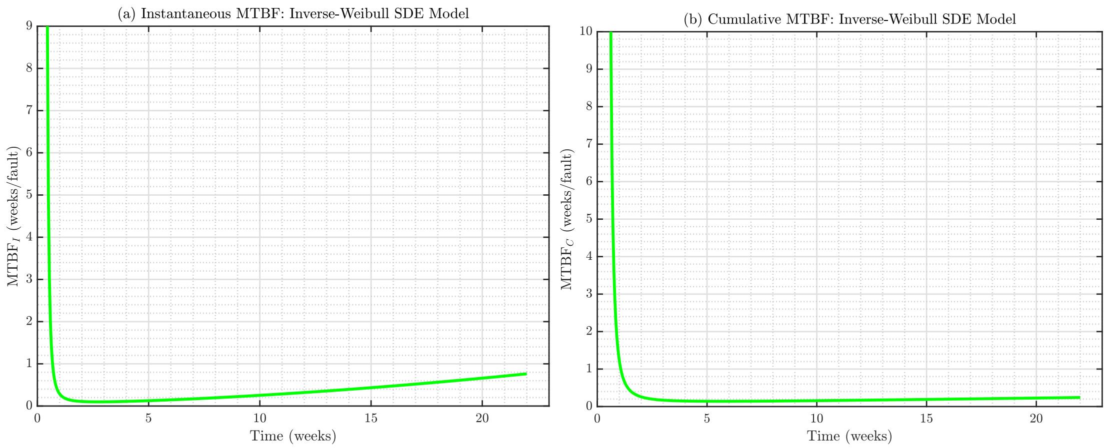

Figure 4: Evolution of Instantaneous (a) and Cumulative (b) MTBF.

Table 3: Estimated Reliability Metrics at Week $t = 22$.

Model	**Expected Faults** $E[N(t)]$	**Failure Rate** $r(t)$ (faults/wk)	**MTBF (weeks)**	
			Instantaneous	Cumulative
Proposed (SDE-IW)	**91.11**	**1.31**	**0.763**	**0.241**
Exponential (GO)	106.17	2.58	0.387	0.207
S-shaped (Yamada)	97.29	1.13	0.888	0.226

10. Conclusion

In this paper, we addressed the limitations of deterministic Software Reliability Growth Models (SRGMs) in capturing the irregular and bursty fault-detection patterns inherent to Open Source Software (OSS) development, where we developed a flexible framework that accounts for environmental noise and uncertain testing dynamics by modeling the fault-detection intensity as a stochastic process governed by a Stochastic Differential Equation (SDE) with an Inverse-Weibull baseline.

Next, in order to validate the proposed model (SDE-IW), we used fault data from the Apache open-source project, and the comparative analysis against classical Non-Homogeneous Poisson Process (NHPP) models (including Goel-Okumoto and Yamada S-shaped models) demonstrated that the SDE-based approach achieves goodness-of-fit, as evidenced by the lowest AIC and BIC scores. Also, the SDE-IW model accurately tracked the stochastic fluctuations in the fault discovery rate, a feature that deterministic models consistently failed to capture. These results substantiate the hypothesis that the fault-count process is best modeled as a Cox process (doubly stochastic Poisson process), in which the intensity is subject to random perturbations driven by Brownian motion.

Finally, this study provides a mathematically effective tool for reliability engineers managing complex software systems. While the current focus remains on software, the probabilistic nature of this framework paves the way for holistic system reliability modeling. For future works, we suggest integrating this SDE-driven software model with Markovian frameworks for hardware standby redundancy, ultimately aiming for a unified dynamic model for mixed hardware-software systems (e.g., standby configurations [29,30,31,32]).

References

1. A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and Mozilla," *ACM Trans. Softw. Eng. Methodol.*, vol. 11, no. 3, pp. 309–346, 2002. https://doi.org/10.1145/567793.567795

2. M. Zhu and H. Pham, “A multi-release software reliability modeling for open source software incorporating dependent fault detection process,” *Ann Oper Res.*, vol. 269, pp. 773–790, 2018. https://doi.org/10.1007/s10479-017-2556-6

3. M. Zhu, "Software reliability modeling and its applications considering fault dependency and environmental factors," PhD Thesis, Rutgers, The State University of New Jersey, 2018. https://doi.org/doi:10.7282/t3-mv83-af31

4. S. Chatterjee, B. Chaudhuri, and C. Bhar, “Optimal release time determination in intuitionistic fuzzy environment involving randomized cost budget for SDE-based software reliability growth model,” *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 2721–2741, 2020. https://doi.org/10.1007/s13369-019-04128-7

5. Z. Zheng, J. Yang, Y. Hu, and X. Wang, “Open-source software reliability modeling with stochastic impulsive differential equations,” *Eksploatacja i Niezawodnosc – Maint. Reliab.*, vol. 25, no. 2, p. 166342, 2023. https://doi.org/10.17531/ein/166342

6. H. Pham, “Software Reliability Models: A Review. In: Pham, H. (eds) Analytics Modeling in Reliability and Machine Learning and Its Applications,” *Springer Series in Reliability Engineering*, Springer, Cham, 2025. https://doi.org/10.1007/978-3-031-72636-1_17

7. A. L. Goel and K. Okumoto, “Time-dependent error-detection rate model for software reliability and other performance measures,” *IEEE Trans. Reliab.*, vol. 28, no. 3, pp. 206–211, 1979. https://doi.org/10.1109/TR.1979.5220566

8. S. Yamada, M. Ohba, and S. Osaki, “S-shaped software reliability growth models and their applications,” *IEEE Trans. Reliab.*, vol. 33, no. 4, pp. 289–292, 1984. https://doi.org/10.1109/TR.1984.5221839

9. P. K. Kapur, H. Pham, A. Gupta and P.C. Jha, *Software Reliability Growth Models. In: Software Reliability Assessment with OR Applications,* Springer Series in Reliability Engineering, Springer, London, 2011. https://doi.org/10.1007/978-0-85729-204-9_2

10. S. Yamada, A Nishigaki and M. Kimura, “A Stochastic Differential Equation Model for Software Reliability Assessment and Its Goodness-of-Fit,” *International Journal of Reliability and Applications*, vol. 4, no. 1, pp. 1–11, 2003.

11. Y. Tamura and S. Yamada, “Optimisation analysis for reliability assessment based on stochastic differential equation modelling for open source software,” *Int. J. Syst. Sci.*, vol. 40, no. 4, pp. 429–438, 2009. https://doi.org/10.1080/00207720802556245

12. C. H. Lee, Y. T. Kim, and D. H. Park, “S-shaped software reliability growth models derived from stochastic differential equations,” *IIE Trans.*, vol. 36, no. 12, pp. 1193–1199, 2004. https://doi.org/10.1080/07408170490507792

13. A. F. Mahdi, H. K. Asker, and I. R. Al-Saiq, “A Deterministic Mathematical Dynamic System Based on the PSITPS Model for Modeling the COVID-19 Epidemic,” *Mathematical Modelling of Engineering Problems*, vol. 11, no. 4, pp. 872–882, 2024. https://doi.org/10.18280/mmep.110404

14. T. Pham and H. Pham, “A generalized software reliability model with stochastic fault-detection rate,” *Ann. Oper. Res.*, vol. 277, no. 1, pp. 83–93, 2019. https://doi.org/10.1007/s10479-017-2486-3

15. J. Yang, M. Ding, M. He, Z. Zheng, and N. Yang, “SDE-based software reliability additive models with masked data using ELS algorithm,” *Journal of King Saud University - Computer and Information Sciences*, vol. 36, p. 101978, 2024. https://doi.org/10.1016/j.jksuci.2024.101978

16. M. Zhao, “Superposition of power-law models for hardware/software system reliability data analysis,” *Int. J. Reliab. Qual. Saf. Eng.*, vol. 10, no. 2, pp. 121–130, 2003. https://doi.org/10.1142/S0218539303001044

17. H. K. Asker and M. S. Kadhim, “T-Periodic Solution of Neutral Stochastic Functional Differential Equations with Infinite Delay,” *Basrah Journal of Science*, vol. 37, no. 2, pp. 181–191, 2019. https://doi.org/10.29072/basjs.20190203

18. H. K. Asker, “Well-posedness and Exponential Estimates for the Solutions to Neutral Stochastic Functional Differential Equations with Infinite Delay,” *Journal of Systems Science and Information*, vol. 8, pp. 387–400, 2020. https://doi.org/10.21078/JSSI-2020-434-13

19. H. K. Asker, “Stability in distribution of neutral stochastic functional differential equations with infinite delay,” *Computational Methods for Differential Equations (CMDE)*, vol. 9, no. 2, pp. 604–622, 2021. https://doi.org/10.22034/cmde.2020.32804.1525

20. B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, 6th ed. Springer, 2013.

21. J. F. C. Kingman, *Poisson Processes.* (Vol. 3), Clarendon Press, 1992.

22. D. R. Cox, “Some statistical methods connected with series of events,” *J. R. Stat. Soc. Ser. B*, vol. 17, no. 2, pp. 129–157, 1955. https://doi.org/10.1111/j.2517-6161.1955.tb00188.x

23. D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*, 2nd ed. Springer, 2003.

24. D. Böhning, *Computer-Assisted Analysis of Mixtures and Applications: Meta-Analysis, Disease Mapping and Others*. Chapman and Hall/CRC, 1999.

25. G. S. Mudholkar, D. K. Srivastava, and M. Freimer, “The exponentiated Weibull family: A reanalysis of the bus-motor-failure data,” *Technometrics*, vol. 37, no. 4, pp. 436–445, 1995. https://doi.org/10.2307/1269735

26. K. P. Burnham and D. R. Anderson, “Multimodel inference: Understanding AIC and BIC in model selection,” *Sociol. Methods Res.*, vol. 33, no. 2, pp. 261–304, 2004. https://doi.org/10.1177/0049124104268644

27. Q. H. Vuong, “Likelihood ratio tests for model selection and non-nested hypotheses,” *Econometrica*, vol. 57, no. 2, pp. 307–333, 1989. https://doi.org/10.2307/1912557

28. J. Yang, M. Zhao, and J. Chen, “ELS algorithm for estimating open source software reliability with masked data considering both fault detection and correction processes,” *Commun. Stat. – Theory Methods*, vol. 51, no. 19, pp. 6792–6817, 2022. https://doi.org/10.1080/03610926.2020.1866610

29. L. Hindi and H. K. Asker, “Analyzing the Impact of Repairable 1-out-of-3 Cold Standby Components on System Availability: A Capacity Analysis,” *Mathematical Modelling of Engineering Problems*, vol. 10, no. 3, pp. 937–942, 2023. https://doi.org/10.18280/mmep.100325

30. L. Hindi and H. K. Asker, “Role of individual component failure in the performance of a 1-out-of-3 cold standby system: A Markov model approach,” *Open Engineering*, vol. 14, no. 1, art. 20220517, 2024. https://doi.org/10.1515/eng-2022-0517

31. H. K. Asker, L. Hindi, and M. S. M. Zabiba, “Individual Component Failures Role in an Imperfect One-out-of-Three Cold and Warm Standby System,” *Boletim da Sociedade Paranaense de Matemática*, vol. 43, no. 4, pp. 1–11, 2025. https://doi.org/10.5269/bspm.78936

32. H. K. Asker, M. T. Kahnger, L. Hindi, and H. K. Ismail, “Absorbing Analysis of a One-Out-of-Three Cold-Warm Standby Cloud Server System with Repairable Switch,” *Mathematical Modelling of Engineering Problems*, vol. 12, no. 11, 2025. https://doi.org/10.18280/mmep.121111

33. M. R. Lyu, *Handbook of Software Reliability Engineering*. McGraw-Hill, Inc., Hightstown, NJ, USA, 1996.

34. A. Mockus, R. T. Fielding and J. Herbsleb, ”A case study of open source software development: the Apache server,” *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium, Limerick, Ireland*, 2000, pp. 263-272. https://doi:10.1145/337180.337209

35. S. Yamada, M. Ohba, and S. Osaki, “S-shaped reliability growth modeling for software error detection,” *IEEE Transactions on Reliability*, vol. R-32, no. 5, pp. 475–484, 1983. https://doi:10.1109/TR.1983.5221735

A. Appendix A: Data set

The data set was obtained from the Apache OSS project and covers the period from September 2010 to July 2012. Table 4 describes the collected data. In Table 4, the “Failures” column displays the total number of software problems discovered between time units t_{i-1} and t_i, while the “Cumulative Failures” column shows the total number of software failures over time t_i [3].

Table 4: Failure data from Apache OSS Project.

Time (t_i)	Failures (d_i)	Cumulative ($N(t_i)$)	Time (t_i)	Failures (d_i)	Cumulative ($N(t_i)$)
1	6	6	12	4	66
2	6	12	13	0	66
3	6	18	14	4	70
4	8	26	15	5	75
5	13	39	16	5	80
6	6	45	17	2	82
7	8	53	18	10	92
8	2	55	19	1	93
9	3	58	20	1	94
10	3	61	21	2	96
11	1	62	22	0	96

B. Appendix B: Derivation of the SDE

B.1. Phenomenological Justification

In classical models, the detection rate is $b(t)$. We posit that stochastic fluctuations in the debugging process are driven by the *magnitude of the remaining work*. Specifically, the noise intensity is proportional to the number of remaining faults $(a - m(t))$. This yields the SDE:

$$dm(t) = b(t)(a - m(t))dt + \sigma(a - m(t))dW(t). \tag{B.1}$$

This formulation avoids the singularity associated with rate-proportional noise (where noise would scale with $b(t)$) and ensures that as the software matures $(m(t) \to a)$, the variance naturally decays to zero.

B.2. Formal Stochastic Calculus Derivation

Consider the detection process modeled as a deterministic drift plus a diffusion term. If one were to start from a Stratonovich SDE of the form

$$dm(t) = (a - m(t))\left[\, b(t)\, dt + \sigma \circ dW(t) \,\right],$$

conversion to Itô form would introduce a drift correction term $-\frac{1}{2}\sigma^2(a - m(t))\, dt$. However, in reliability modeling, the Itô interpretation is standard because it reflects non-anticipating (causal) debugging effort. We therefore posit the Itô SDE (2.1) directly, which is empirically and theoretically justified (Remark 2.1).

C. Appendix C: Probability Density of $m(t)$

Given the explicit solution (3.1) for the SDE (2.1):

$$m(t) = a\left[1 - \exp\left(-\int_0^t b(s)\, ds - \sigma W(t) - \tfrac{1}{2}\sigma^2 t\right)\right], \qquad \text{(C.1)}$$

we derive the transition probability density $f_{m(t)|m(s)}(x \mid y)$ for $0 \le s < t$ and $0 \le y < x < a$.

C.1. Conditional Distribution Given $m(s) = y$

From (C.1), the increment $W(t) - W(s) \sim \mathcal{N}(0, t - s)$ is independent of $\mathcal{F}_s$. Solving (C.1) for the Wiener increment yields:

$$\sigma\big(W(t) - W(s)\big) = \ln\left(\frac{a - y}{a - x}\right) - \int_s^t b(u)\, du - \tfrac{1}{2}\sigma^2(t - s). \qquad \text{(C.2)}$$

Since $W(t) - W(s) \sim \mathcal{N}(0, t - s)$, its density is:

$$f_{W(t)-W(s)}(z) = \frac{1}{\sqrt{2\pi(t - s)}} \exp\left(-\frac{z^2}{2(t - s)}\right). \qquad \text{(C.3)}$$

Applying the change-of-variables formula with Jacobian:

$$\left|\frac{d}{dx}\left[W(t) - W(s)\right]\right| = \frac{1}{\sigma(a - x)}, \qquad \text{(C.4)}$$

we obtain the transition density:

$$f_{m(t)|m(s)}(x \mid y) = \frac{1}{\sigma(a - x)\sqrt{2\pi(t - s)}} \exp\left(-\frac{\left[\ln\left(\frac{a-y}{a-x}\right) - \int_s^t b(u)\, du\right]^2}{2\sigma^2(t - s)}\right). \qquad \text{(C.5)}$$

Remark C.1 (Log-Normal Structure) Equation (C.5) confirms that $a - m(t) \mid m(s) = y$ follows a log-normal distribution:

$$a - m(t) \mid m(s) = y \sim \text{Lognormal}\left(\mu, \sigma^2(t - s)\right), \quad \mu = \ln(a - y) - \int_s^t b(u)\, du. \qquad \text{(C.6)}$$

This is consistent with the multiplicative noise structure in (2.1), as geometric Brownian motion (the solution to $dX = \mu X dt + \sigma X dW$) has log-normal marginals [20].

C.2. Unconditional Density and Normalization

The unconditional density $f_{m(t)}(x)$ is obtained by setting $s = 0$, $y = m(0) = 0$ in (C.5):

$$f_{m(t)}(x) = \frac{1}{\sigma(a-x)\sqrt{2\pi t}} \exp\left(-\frac{\left[\ln\left(\frac{a}{a-x}\right) - \int_0^t b(u)\,du\right]^2}{2\sigma^2 t} \right), \quad 0 < x < a. \tag{C.7}$$

To verify normalization, substitute $z = \ln\left(\frac{a}{a-x}\right) - \int_0^t b(u)\,du$, so that $x = a\left[1 - e^{-(z+\int_0^t b)}\right]$ and $dx = ae^{-(z+\int_0^t b)}dz$. Then:

$$\int_0^a f_{m(t)}(x)\,dx = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi t}} \exp\left(-\frac{z^2}{2\sigma^2 t}\right) dz = 1, \tag{C.8}$$

confirming that (C.7) is a valid probability density.

C.3. Implication for Fault Count Distribution

Crucially, while $m(t)$ has a continuous density (C.7), the fault count $N(t)$ remains discrete. Its unconditional distribution is a *Poisson–lognormal mixture*:

$$\mathbb{P}\big(N(t) = k\big) = \int_0^a \frac{x^k}{k!} e^{-x} f_{m(t)}(x)\,dx, \quad k = 0, 1, 2, \ldots \tag{C.9}$$

This explains why the likelihood function in Section 6 must be based on (C.9) rather than a continuous density for $N(t)$—a common oversight in prior SDE-based SRGMs [12].

D. Appendix D: Improper Integration of Inverse-Weibull Hazard

The cumulative intensity function for the Inverse-Weibull hazard rate (5.2) requires careful treatment due to the singularity at $t = 0$. We provide a rigorous derivation of (5.3) and analyze its asymptotic behavior.

D.1. Evaluation of the Improper Integral

Consider the improper integral:

$$I(t) = \int_0^t b(s)\,ds = \lim_{\epsilon\to 0^+} \int_\epsilon^t \frac{\gamma\alpha^\gamma s^{-(\gamma+1)} e^{-(\alpha/s)^\gamma}}{1 - e^{-(\alpha/s)^\gamma}}\,ds. \tag{D.1}$$

Apply the substitution $u = (\alpha/s)^\gamma$, so that $s = \alpha u^{-1/\gamma}$ and $ds = -\frac{\alpha}{\gamma} u^{-(1+1/\gamma)} du$. When $s = \epsilon$, $u = (\alpha/\epsilon)^\gamma \to \infty$; when $s = t$, $u = (\alpha/t)^\gamma$. Thus:

$$\begin{aligned}
I(t) &= \lim_{U\to\infty} \int_{(\alpha/t)^\gamma}^U \frac{\gamma\alpha^\gamma (\alpha u^{-1/\gamma})^{-(\gamma+1)} e^{-u}}{1 - e^{-u}} \cdot \frac{\alpha}{\gamma} u^{-(1+1/\gamma)}\,du \\
&= \lim_{U\to\infty} \int_{(\alpha/t)^\gamma}^U \frac{e^{-u}}{1 - e^{-u}}\,du \\
&= \lim_{U\to\infty} \int_{(\alpha/t)^\gamma}^U \sum_{k=1}^\infty e^{-ku}\,du \quad \text{(geometric series for } u > 0) \\
&= \sum_{k=1}^\infty \lim_{U\to\infty} \int_{(\alpha/t)^\gamma}^U e^{-ku}\,du \\
&= \sum_{k=1}^\infty \frac{1}{k} e^{-k(\alpha/t)^\gamma} \\
&= -\ln\big(1 - e^{-(\alpha/t)^\gamma}\big),
\end{aligned} \tag{D.2}$$

where the final equality follows from the Taylor series $-\ln(1-x) = \sum_{k=1}^{\infty} x^k/k$ for $|x| < 1$. Since $e^{-(\alpha/t)^\gamma} \in (0,1)$ for all $t > 0$, the series converges absolutely.

Remark D.1 (Convergence Conditions) The integral (D.1) converges if and only if $\gamma > 0$. For $\gamma \leq 0$, the integrand behaves as $s^{-(\gamma+1)} \sim s^{-1}$ near $s = 0$, leading to logarithmic divergence. This justifies our assumption $\gamma > 0$ in Section 5.

Fatimah Saad Hameed,
Department of Mathematics,
Faculty of Computer Science and Mathematics, University of Kufa,
Iraq.
E-mail address: fatimas.hajjami@student.uokufa.edu.iq

and

Hussein K. Asker,
Department of Mathematics,
Faculty of Computer Science and Mathematics,
University of Kufa, Iraq.
E-mail address: husseink.askar@uokufa.edu.iq