PERFORMANCE ANALYSIS OF SOFTWARE SYSTEM SUBJECT TO PREVENTIVE MAINTENANCE AND SOFTWARE UP-GRADATION FACILITY

Kavita Thukral[1] and Shiv Kumar Sharma[2]

•

[1, 2] Department of Mathematics, Chandigarh University, Mohali, Punjab, India
ktkavita540@gmail.com, shivksharma2005@gmail.com

Abstract

Every industrialist as well as scientist try to satisfy the customers requirement by providing reliable software systems and a software system is reliable and profitable if it is easily manageable. This study analyzes the criteria under which software performs correctly with software upgrade and preventive maintenance facility. It is observed that researcher threw light on negligible, major, and load faults cause software to fail, after which load recovery, major, and negligible upgrades are carried out, but there are no discussion on preventive maintenance concept. Sometimes, to enhance the software reliability and performance, preventive maintenance is utilized and it is better to upgrade the software before completely failure. The weibull distributions are followed by all of these failures and upgrade rates. The malfunctioning program can always be fixed by a skilled technician. The software performance is analyzed by applying the regenerative point technique and the semi markov process. Software reliability metrics including mean time to software system failure, software availability, and software profit values are calculated using tables.

Keywords: Software reliability, negligible fault, major fault, load failure, preventive maintenance, regenerative point technique.

I. Introduction

Software is a key component of the majority of software systems. In the rapidly expanding era of digitization, most people now rely significantly on software tools to complete tasks accurately and on schedule. Therefore, in order to satisfy the clients, the software system must perform well. However, it is truth that no system can function indefinitely since software systems and machinery malfunction due to little fault, major fault, and high loads. Software upgrades and preventive maintenance can be employed in certain circumstances to enhance the availability and profit values of software systems.

Over the years, researchers have suggested some reliability improvement techniques using stochastic process with semi markov approach. Smith [20] described the performance parameters of software system under different conditions. Coleman et al. [4] examined the application of software maintainability models in industrial software systems. Huang [5] evaluated the performance of software reliability growth models with testing-effort and change-point. Misra [16] threw light on modeling design and coding factors that analyzed the maintainability of software systems. Kumar and Malik [7] discussed reliability measures of computer systems with priority in repair activities subject to maximum operation and repair times. Malhotra and Chug [14] evaluated the software maintainability with systematic literature review and current trends. The profit analysis of a computer system has been discussed by Kumar and Saini [8] regarding preventive maintenance. Malik et al. [20] examined the reliability of a non series parallel system with weibull failure laws. Barak et al. [1] examined the reliability, availability and profit values of milk industry subject to repair facility. Kavita and Sharma [6] analyzed the estimation of software reliability and performance using lindley distribution. Barak et al. [2] described the effects of refreshment for the technician in a stochastic cold standby system. Kumar and Sharma [9] descried the availability and profit values of stochastic repairable cold standby system under refreshments. Kumar et al. [11] described the reliability measures of hardware and software system using weibull distribution under load failure facility. Kumar et al. [13] analyzed the mean time to system failure, availability of the system, busy period of the system, estimated number of visits made by the technician and profit function under inspection and refreshment facility. Kumar et al. [12] analyzed the reliability measures of a computer system under hardware repair and software upgrade facility. Kumar and Sharma [10] analyzed the reliability measures of repairable cold standby system subject to reboot facility under refreshments. Behera and Agarwal [3] explored the software reliability prediction and management incorporating change points based on testing effort. Rahul et al. [17] analyzed the reliability characteristics of juice industry using stochastic process and regenerative point graphical technique. Reddy et al. [18] examined the reliability characteristics of utensil industry subject to repair facility. Sharma and Rana [19] described the advancing software reliability with time series approach under identification of defects with artificial neural network.

It has been determined that software upgradation improve the machines as well as software systems availability and profitability. Regenerative point technique with weibull distribution has been used to analyze the reliability characteristics such as mean sojourn times, mean time to software system failure, availability, technician busy period due to software upgrades, expected number of technician visits, and profit function. Tables explore these reliability metrics.

II. System Assumptions

There are following software system assumptions

- Initially, it is assumed that the software work properly.
- The software system is timely up-graded to avoid the negligible fault/major fault/ load failure of software.
- Preventive maintenance is performed to enhance the reliability of the software.
- An expert technician is available to up-grade the software.
- After up-grade, the failed software work like a new software.
- Parameters used to describe the software system are all statistically independent.
- Software failure time and up-gradation time follow the weibull distribution.
- Preventive maintenance request time and preventive maintenance time follow weibull distribution.

III. System Notations

To explain the software system, there are following notations

R	Set of regenerative state (S_0, S_1, S_2, S_3).
S_{Op}	Software in operative mode.
S_{Nf}	Software with negligible fault/failure.
S_{Mf}	Software with major fault/failure.
S_{Lf}	Software with load failure.
$q_{ij}(t)$	PDF of the first passage time from a regenerative state i to a regenerative state j or to a failed state j without visiting any other regenerative state in (0,1].
μ_i	Mean sojourn time spent in the state 'i' before visiting any other states.

μ_i'	Total unconditional time spent before transiting to any other regenerative state while the system entered regenerative state '*i*' at t=0.
$M_i(t)$	Probability that the system is originally up in the regenerative state $S_i \in R$ up to at the time (*t*) without passing through any other $S_i \in R$.
$W_i(t)$	The probability that the technician is busy in the state S_i up to time (t) without making any transition to any other $S_i \in R$ or returning to the same via one or more non- regenerative states.
$f_1(t) = \alpha\eta t^{\eta-1}e^{-\alpha t^\eta}$	PDF of the negligible failure time.
$f_2(t) = \beta\eta t^{\eta-1}e^{-\beta t^\eta}$	PDF of the major failure time.
$f_3(t) = \gamma\eta t^{\eta-1}e^{-\gamma t^\eta}$	PDF of the load failure time.
$f_4(t) = \psi\eta t^{\eta-1}e^{-\psi t^\eta}$	PDF of the negligible failure time to major failure time.
$f_4(t) = \xi\eta t^{\eta-1}e^{-\xi t^\eta}$	PDF of the preventive maintenance request time.
$g_1(t) = k\eta t^{\eta-1}e^{-kt^\eta}$	PDF of the negligible up-gradation time.
$g_2(t) = l\eta t^{\eta-1}e^{-lt^\eta}$	PDF of the major repair time.
$g_3(t) = m\eta t^{\eta-1}e^{-mt^\eta}$	PDF of the load repair time.
$g_5(t) = h\eta t^{\eta-1}e^{-ht^\eta}$	PDF of the preventive maintenance time.
○/ ⬠/ ▭/ ●	Upstate/ negligible failed state/ failed state/ regenerative state.

IV. Circuits Descriptions

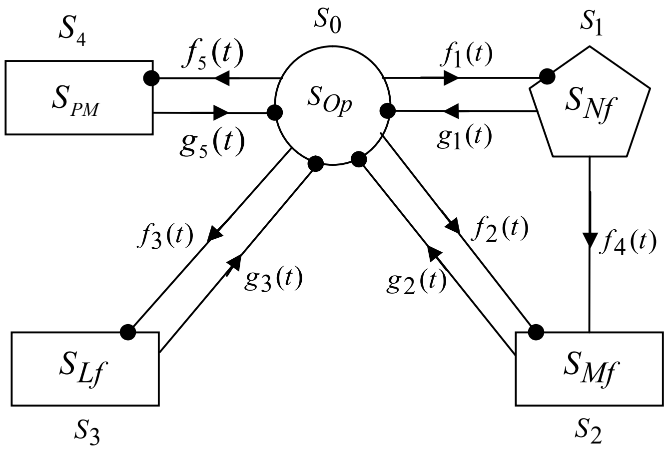

Figure 1 *State Transition Diagram*

V. Transition Probabilities

There are following transition probabilities

$p_{0,1} = \alpha/(\alpha+\beta+\gamma+\xi)$, $p_{0,2} = \beta/(\alpha+\beta+\gamma+\xi)$

$p_{0,3} = \gamma/(\alpha+\beta+\gamma+\xi)$, $p_{0,4} = \xi/(\alpha+\beta+\gamma+\xi)$,

$$p_{1,0} = k/(k+\psi)\ ,\ p_{1,0;2} = \psi/(k+\psi)\ ,\ p_{1,2} = \psi/(k+\psi)\ ,\ p_{2,0} = p_{3,0} = 1 \tag{1}$$

It is also simplified that

$$p_{0,1} + p_{0,2} + p_{0,3} + p_{0,4} = p_{1,0} + p_{1,2} = 1\ ,\ p_{1,0} + p_{1,0;2} = 1 \tag{2}$$

VI. Mean Sojourn Time

For the particular state, it becomes

$$\mu_0 = \frac{\Gamma(1+\frac{1}{\eta})}{(\alpha+\beta+\gamma+\xi)^{1/\eta}},\ \mu_1 = \frac{\Gamma(1+\frac{1}{\eta})}{(k+\psi)^{1/\eta}},\ \mu_2 = \frac{\Gamma(1+\frac{1}{\eta})}{(l)^{1/\eta}},\ \mu_3 = \frac{\Gamma(1+\frac{1}{\eta})}{(m)^{1/\eta}}$$

$$\mu_1' = \Gamma(1+\frac{1}{\eta})\left[\frac{1}{(k+\psi)^{1/\eta}} + \frac{\psi}{l(k+\psi)^{1/\eta}}\right] \tag{3}$$

VII. Evaluation of Parameters

I. Mean Time to System Failure

Let $\varphi_i(t)$ is the cumulative distribution function of the first passage time from $S_i \in R$ to a failed state and treating the failed states as an absorbing states then upcoming recursive relation for $\varphi_i(t)$ becomes

$$\varphi_0(t) = Q_{0,1}(t) \otimes \varphi_1(t) + Q_{0,2}(t) + Q_{0,3}(t) + Q_{0,4}(t)$$
$$\varphi_1(t) = Q_{1,0}(t) \otimes \varphi_0(t) + Q_{1,2}(t)$$
$$\varphi_2(t) = Q_{2,0}(t)$$
$$\varphi_3(t) = Q_{3,0}(t)$$
$$\varphi_4(t) = Q_{4,0}(t) \tag{4}$$

Now taking LST on above equation (4) and solving for $\varphi_0^{**}(s)$, obtained

$$R^*(s) = \frac{1-\varphi_0^{**}(s)}{s} \tag{5}$$

The system reliability is obtained by using inverse LT on the above equation (5), get

$$MTSF = \lim_{s\to 0} \frac{1-\varphi_0^{**}(s)}{s} = \frac{\mu_0 + p_{0,1}\mu_1}{1-p_{0,1}p_{1,0}}$$

$$MTSF = \frac{\Gamma(1+\frac{1}{\eta})\left[\frac{\alpha}{(\alpha+\beta+\gamma+\xi)(k+\psi)^{1/\eta}} + \frac{1}{(\alpha+\beta+\gamma+\xi)^{1/\eta}}\right]}{\left[1-\frac{\alpha k}{(\alpha+\beta+\gamma+\xi)(k+\psi)}\right]} \tag{6}$$

II. Software Availability

Let $A_i(t)$ is the probability that software system is in up-state at the time 't' specified that the system arrives at the regenerative-state S_i at $t = 0$ and then upcoming recursive relation for $A_i(t)$ becomes

$$A_0(t) = M_0(t) + q_{0,1}(t) \oplus A_1(t) + q_{0,2}(t) \oplus A_2(t) + q_{0,3}(t) \oplus A_3(t) + q_{0,4}(t) \oplus A_4(t)$$
$$A_1(t) = M_1(t) + [q_{1,0}(t) + q_{1,0;2}(t)] \oplus A_0(t)$$
$$A_2(t) = q_{2,0}(t) \oplus A_0(t)$$
$$A_3(t) = q_{3,0}(t) \oplus A_0(t)$$
$$A_4(t) = q_{4,0}(t) \oplus A_0(t) \tag{7}$$

where, $M_0(t) = e^{-(\alpha+\beta+\gamma+\xi)t^n}$, $M_1(t) = e^{-(k+\psi)t^n}$

Using LT on equation (7) and solving for $A_0^*(s)$, then steady-state availability is given by

$$A_0 = \lim_{s\to 0} sA_0^*(s) = \frac{N_A}{D'} = \frac{\mu_0 + p_{0,1}\mu_1}{\mu_0 + p_{0,1}\mu_1' + p_{0,2}\mu_2 + p_{0,3}\mu_3 + p_{0,4}\mu_4} \tag{8}$$

where, $$N_A = \Gamma(1+\frac{1}{\eta})\left[\frac{1}{(\alpha+\beta+\gamma+\xi)^{1/\eta}} + \frac{\psi}{(\alpha+\beta+\gamma+\xi)(k+\psi)^{1/\eta}}\right] \tag{9}$$

$$D' = \Gamma(1+\frac{1}{\eta})\left[\begin{array}{l}\left\{\begin{array}{l}\frac{1}{(\alpha+\beta+\gamma+\xi)^{1/\eta}} + \frac{\beta}{(\alpha+\beta+\gamma+\xi)(l)^{1/\eta}} \\ + \frac{\gamma}{(\alpha+\beta+\gamma+\xi)(m)^{1/\eta}} + \frac{\xi}{(\alpha+\beta+\gamma+\xi)(h)^{1/\eta}}\end{array}\right\} + \\ \frac{\alpha}{(\alpha+\beta+\gamma+\xi)}\left\{\frac{1}{(k+\xi)^{1/\eta}} + \frac{\psi}{l(k+\psi)^{1/\eta}}\right\}\end{array}\right] \tag{10}$$

III. Busy Period of the Technician

Let $B_i(t)$ is the probability that the technician is busy at time 't' specified that the system arrives at the S_i at $t = 0$. Then the upcoming recursive relation for $B_i(t)$ obtained as

$$B_0(t) = q_{0,1}(t) \oplus B_1(t) + q_{0,2}(t) \oplus B_2(t) + q_{0,3}(t) \oplus B_3(t) + q_{0,4}(t) \oplus B_4(t)$$
$$B_1(t) = W_1(t) + [q_{1,0}(t) + q_{1,0;2}(t)] \oplus B_0(t)$$
$$B_2(t) = W_2(t) + q_{2,0}(t) \oplus B_0(t)$$
$$B_3(t) = W_3(t) + q_{3,0}(t) \oplus B_0(t)$$
$$B_4(t) = W_4(t) + q_{4,0}(t) \oplus B_0(t) \tag{11}$$

where, $W_1(t) = \overline{F}_4(t) \oplus \overline{G}_1(t)$, $W_2(t) = \overline{G}_2(t)$, $W_3(t) = \overline{G}_3(t)$ and $W_4(t) = \overline{G}_5(t)$

Using LT on equation (11) and solving for $B_0^*(s)$ then

$$B_0 = \lim_{s\to 0} sB_0^*(s) = \frac{N_B}{D'} = \frac{p_{0,1}\mu_1' + p_{0,2}\mu_2 + p_{0,3}\mu_3}{\mu_0 + p_{0,1}\mu_1' + p_{0,2}\mu_2 + p_{0,3}\mu_3 + p_{0,4}\mu_4} \tag{12}$$

where, $$N_B = \Gamma(1+\frac{1}{\eta})\left[\begin{array}{l}\left\{\frac{\beta}{(\alpha+\beta+\gamma+\xi)(l)^{1/\eta}} + \frac{\gamma}{(\alpha+\beta+\gamma+\xi)(m)^{1/\eta}}\right\} + \\ \frac{\alpha}{(\alpha+\beta+\gamma+\xi)}\left\{\frac{1}{(\alpha+\beta+\gamma+\xi)^{1/\eta}} + \frac{\psi}{(\alpha+\beta+\gamma+\xi)(k+\psi)^{1/\eta}}\right\}\end{array}\right] \tag{13}$$

and D' is defined earlier in equation (10).

IV. Estimated number of visits made by the Technician

Let $N_i(t)$ is the estimated number of visits by the technician for software up-gradation in (0, t] specified that technician arrives $S_i \in R$ at t=0. The upcoming recursive relation for $N_i(t)$ becomes

$$N_0(t) = Q_{0,1}(t) \otimes [1 + N_1(t)] + Q_{0,2}(t) \otimes [1 + N_2(t)] + Q_{0,3}(t) \otimes [1 + N_3(t)] + Q_{0,4}(t) \otimes [1 + N_4(t)]$$
$$N_1(t) = [Q_{1,0}(t) + Q_{1,0;2}(t)] \otimes N_0(t)$$
$$N_2(t) = Q_{2,0}(t) \otimes N_0(t)$$
$$N_3(t) = Q_{3,0}(t) \otimes N_0(t)$$
$$N_3(t) = Q_{4,0}(t) \otimes N_0(t) \tag{14}$$

Using LST on equation (14) and solving for $N_0^{**}(s)$. Then

$N_0 = \lim_{s\to 0} sN_0^{**}(s) = \frac{N_r}{D'}$, where $N_r = 1$

and D' is defined earlier in equation (10).

V. Profit Analysis

The profit analysis of the system can be evaluated by using the profit function

$$P = E_0 A_0 - E_1 B_0 - E_2 N_0 \tag{15}$$

Where, E_0 = 5000 (Revenue per unit uptime of the system is available)

E_1 =300 (Charge per unit time for which technician is busy in software up-grade)

E_2 = 100 (Charge per unit visit made by the technician for the defined purpose)

VIII. Discussion

Here, preventive maintenance rate play an important role in software reliability and performance evaluation. To calculate the reliability measures such that mean time to software system failure, software availability, and profit values are calculated by using particular cases of failure time and up-gradation time such as $\alpha = \beta = \gamma = \psi = \xi = \alpha_1$ and $k = l = m = h = k_1$ respectively. Here, we have used the shape parameter η=0.5 for the reliability measures.

From table-1 it is clarified that mean time to software system failure values decreases with the increment of software failure rate (α_1) but it is enhance with increment in software up-gradation rate (k_1) from 2 to 2.5, 3, 3.5, & 4 respectively.

Table 1: *MTSF vs. Software Failure Rate* (α_1)

(α_1) ↓	k_1=2, η=0.5	k_1=2.5, η=0.5	k_1=3, η=0.5	k_1=3.5, η=0.5	k_1=4, η=0.5
0.03	51.72014	62.18437	69.09374	79.42516	86.55808
0.035	48.62661	57.35737	63.73042	73.25431	79.84645
0.04	45.87798	53.21386	59.12651	67.95793	74.08405
0.045	43.41996	49.61988	55.13325	63.36451	69.08506
0.05	41.20925	46.47405	51.63784	59.34428	64.70879
0.055	39.21056	43.69831	48.55367	55.79732	60.84685
0.06	37.39508	41.23157	45.81285	52.64549	57.41445
0.065	35.73288	39.02541	43.36157	49.82682	54.34432
0.07	34.22656	37.04096	41.15662	47.29159	51.58246
0.075	32.82775	35.24665	39.16294	44.99854	49.08502

Table 2: *Availability vs. Software Failure Rate* (α_1)

(α_1) ↓	k_1=2, η=0.5	k_1=2.5, η=0.5	k_1=3, η=0.5	k_1=3.5, η=0.5	k_1=4, η=0.5
0.03	0.942288	0.958545	0.966458	0.970756	0.980568
0.035	0.939973	0.956109	0.964075	0.968365	0.978178
0.04	0.937661	0.953672	0.961704	0.965976	0.976241
0.045	0.935358	0.951233	0.959333	0.963594	0.974239
0.05	0.933036	0.948788	0.956965	0.961216	0.972568
0.055	0.930717	0.946336	0.954582	0.958827	0.978622
0.06	0.928392	0.943874	0.952197	0.956437	0.967798
0.065	0.926059	0.941402	0.949804	0.954046	0.965492
0.07	0.923716	0.938917	0.947401	0.951634	0.963101
0.075	0.921362	0.936419	0.944987	0.949217	0.960628

Table 4: *Profit vs. Software Failure Rate* (α_1)

(α_1) ↓	k_1=2, η=0.5	k_1=2.5, η=0.5	k_1=3, η=0.5	k_1=3.5, η=0.5	k_1=4, η=0.5
0.03	4710.33	4791.29	4831.37	4852.87	4898.38
0.035	4698.61	4778.95	4819.76	4840.85	4889.88
0.04	4686.89	4766.68	4807.39	4828.74	4880.07
0.045	4675.16	4754.23	4795.46	4816.78	4869.92
0.05	4663.43	4741.82	4783.38	4804.64	4859.42
0.055	4651.65	4729.37	4771.34	4792.56	4848.54
0.06	4639.83	4716.86	4759.26	4780.44	4837.25
0.065	4627.96	4704.28	4747.18	4768.28	4825.56
0.07	4616.58	4691.64	4734.93	4756.77	4816.28
0.075	4604.56	4678.92	4722.67	4743.86	4802.83

Table-2 threw light on the availability of the software and it is decreases with increment of software failure rate (α_1). Also, it is analyzed that availability of software enhance with increment in software up-gradation rate (k_1) from 2 to 2.5, 3, 3.5, & 4 respectively.

Profit is the essential parameter of reliability that inform customers or industrialist whether the software system is in their favour or not. Software are used in the computing machine and well as to handle the robotic systems in particular manners. Sometimes, it is observed that preventive maintenance enhance these reliability measures.

Profit values are calculated from table-3, and it is observed that it decreases with increment of software failure rate (α_1) but it is enhances with increment in software up-gradation rate(k_1) from 2 to 2.5, 3, 3.5, & 4 respectively.

IX. Conclusion

Mean time to software system failure, availability, and profit function are essential parameters of reliability measure that inform customers or industrialist whether the software system is in their favour or not. Software are used in the computing machine and well as to handle the robotic systems in particular manners. Sometimes, it is observed that preventive maintenance enhance these reliability measures. The performance of the software is discussed using the regenerative point technique. The above tables explore that when the upgradation rate increases then the MTSF, availability and profit values also increase but when the failure rate increases then the MTSF, availability and profit values decrease.

X. Future Scope

It is observed that the role of the regenerative point technique with preventive maintenance for the software system will be beneficial and also used by the management, manufacturers and the persons engaged in reliability engineering and working on analyzing the nature and performance of the computing system like robot and automatic electric machine.

References

[1] Barak, M. S., Garg, R., and Kumar, A. (2021). Reliability measures analysis of a milk plant using RPGT. *Life Cycle Reliability and Safety Engineering*, *10*(3), 295-302.

[2] Barak, M. S., Kumar, A., and Garg, R. (2023). Contribution of Refreshment Provided to the Server During His Job in the Repairable Cold Standby System. *Computational Intelligence in Sustainable Reliability Engineering*, 251-267.

[3] Behera, A. K., and Agarwal, P. (2024). A software reliability prediction and management incorporating change points based on testing effort. *Reliability: Theory and Applications*, *19*(2(78)), 91-100.

[4] Coleman, D., Lowther, B., and Oman, P. (1995). The application of software maintainability models in industrial software systems. *Journal of Systems and Software*, *29*(1), 3-16.

[5] Huang, C. Y. (2005). Performance analysis of software reliability growth models with testing-effort and change-point. *Journal of Systems and Software*, *76*(2), 181-194.

[6] Kavita, and Sharma, S. K. (2022, August). Estimation of software reliability using Lindley distribution based on MLE and UMVUE. In *International workshop of Mathematical Modelling, Applied Analysis and Computation* (pp. 299-316). Cham: Springer Nature Switzerland.

[7] Kumar, A., and Malik, S. C. (2011). Profit analysis of a computer system with priority to so- ftware replacement over hardware repair subject to maximum operation and repair time. *International Journal of Engineering Science and Technology (IJEST)*, *3*(10), 7452-7468.

[8] Kumar, A., and Saini, M. (2018). Profit analysis of a computer system with preventive mai- ntenance and priority subject to maximum operation and repair times. Iran *Journal of Computer Science*, *1*(3), 147-153.

[9] Kumar, A., and Sharma, A. (2023). Profit analysis of repairable cold standby system under refreshment. *Reliability Theory and Applications*, *18*(4), 604-612.

[10] Kumar, A., and Sharma, A. (2024). Profit analysis of repairable cold standby system subject to reboot facility under. *Reliability: Theory and Applications*, *19*(2 (78)), 594-603.

[11] Kumar, A., Garg, R., and Barak, M. S. (2023). Performance analysis of computer systems with Weibull distribution subject to software upgrade and load recovery. *Life Cycle Reliability and Safety Engineering*, *12*(1), 51-63.

[12] Kumar, A., Garg, R., and Barak, M. S. (2023). Performance of computer system with three types of failure using weibull distribution subject to hardware repair and software up gradation. *International Journal of System Assurance Engineering and Management*, *14*(1), 483-491.

[13] Kumar, A., Garg, R., and Barak, M. S. (2023). Reliability measures of a cold standby system subject to refreshment. *International Journal of System Assurance Engineering and Management*, *14*(1), 147-155.

[14] Malhotra, R., and Chug, A. (2016). Software maintainability: Systematic literature review and current trends. *International Journal of Software Engineering and Knowledge Engineering*, *26*(08), 1221-1253.

[15] Malik, S. C., Chauhan, S. K., and Ahlawat, N. (2020). Reliability analysis of a non series parallel system of seven components with Weibull failure laws. *International Journal of System Assurance Engineering and Management*, *11*(3), 577-582.

[16] Misra, S. C. (2005). Modeling design/coding factors that drive maintainability of software systems. *Software Quality Journal*, *13*, 297-320.

[17] Rahul, Yadav, M., and Kumar, H. (2024). Profit analysis of repairable juice plant. *Reliability: Theory and Applications*, *19*(1 (77)), 688-695.

[18] Reddy, H. I., Yadav, M., and Kumar, H. (2024). Stochastic analysis of the utensil industry subject to repair facility. *Reliability: Theory and Applications*, *19*(2 (78)), 170-177.

[19] Sharma, S. K., and Rana, R. K. (2024). Advancing software reliability with time series insights: A non-autoregressive ANN approach. *Quality and Reliability Engineering International*, *40*(8), 4166-4186.

[20] Smith, J. (1991). Performance engineering of software systems. *Journal of the Operation Research Society*, *42*(10), 903-904.