



Different Ranking of NHPP Software Reliability Growth Models with Generalised Measure and Predictability

Nguyen Hung-Cuong

School of Information and Communication Technology
Hanoi University of Science and Technology

Huynh Quyet-Thang

School of Information and Communication Technology
Hanoi University of Science and Technology

Le Hai-Trieu

School of Information and Communication Technology
Hanoi University of Science and Technology

ABSTRACT

In recent years, the authors have proposed a number of Non-homogeneous Poisson process (NHPP) software reliability growth models (SRGMs) to analyse and measure the growth of software reliability during production process. This study works with a mathematics methodology to evaluate and then rank some basic NHPP SRGMs. Characteristics of ranking are: the fit of calculated occurrence failure times with real occurrence failure times; and the fit of a predicted time of the next failure with a real one. From a set of individual measures, the methodology mentioned will be deployed. The implementation of some SRGMs with several real data sets confirms that the ranking of SRGMs depends on the data sets.

General Terms:

Software Engineering

Keywords:

Software Reliability Modelling, Non-homogeneous Poisson Process, Probability Distribution

1. INTRODUCTION

Software permeates our daily life. There is probably no other human-made material which is more omnipresent than software in our modern society. It has become a crucial part of many aspects of the society: home appliances, telecommunications, auto-mobiles, air-planes, shopping, auditing, web teaching, personal entertainment, and so on. In particular, science and technology demand high-quality software for making improvements and breakthroughs. However, as the functionality of computer operations becomes more essential and more complicated and critical applications increase in size and complexity, there is a great need for looking at ways to qualify and predict the reliability of computer systems in various complex operating environments.

The challenges in software reliability studies not only stem from the size, complexity, difficulty, and novelty of software applications in various domains. It also relates to the knowledge, training, experience and characters of the software engineers involved. There are many techniques to work with the reliability measure of software system. One of the most famous techniques is using NHPP to model the occurrence time of failure probabilistically [1, 2, 3, 4]. Despite the fact that software reliability models based on NHPP have become successful tools in software reliability engineering, not all of these models can be adapted and work effectively on a specific problem. This problem can occur when maximum likelihood estimation (MLE) system of equations can not be solved. Therefore, selecting an optimal SRGM to use in a particular case has been an interesting area for researchers in domain of software reliability.

There are many ways to evaluate an object. Any measuring methodologies have its advantages and limitations. There are some following measures: mean square error, mean absolute error, mean difference error, accuracy of estimation, etc.. Sometimes, practitioner can not choose a good measure in a specific case. So the authors introduce some mathematics procedure to generate a generalised measure based on all of those.

From the need to rank SRGMs and an idea about generalised measure mentioned above, the authors apply an mathematics technique to calculate this measure based on real data sets. Based on this result, some SRGMs will be ranked with two characteristics: the fit of calculated occurrence failure times with real occurrence failure times; and the fit of the predicted time of next failure with the real one. Our paper's organization is as follows: after this introduction section, section 2 describes a technique for using NHPP in SRGMs. Section 3 introduces ranking methodology and section 4 shows some experimental result. The last sections summaries our work and provides some discussion on extended ideas.

2. DESCRIPTION OF NHPP SRGMS

Let examine about NHPP SRGMs, one of the most basic techniques in reliability theory of software system.

2.1 Non-homogeneous Poisson process (NHPP)

In probability theory, a stochastic process, known as random process (widely used) is a collection of random variables; this is often used to represent the evolution of some random value, or system, over time. Let have some basic definitions as follows [5]:

Definition 1. Counting process

A stochastic process $\{N(t), t > 0\}$ is called a *counting process* if $N(t)$ denotes the total number of events that occurred up to time t with properties:

- $N(t) \geq 0, \forall t \geq 0$.
- $N(t)$ is an integer.
- $N(t)$ is non-decreasing: if $t_1 \leq t_2$ then $N(t_1) \leq N(t_2)$.
- If $t_1 < t_2$, then $N(t_2) - N(t_1)$ is the number of events occurred during the interval $(t_1, t_2]$.

In probability theory, a Poisson process is a stochastic process that counts the number of events.

Definition 2. Poisson process

A counting process $\{N(t), t \geq 0\}$ is called a Poisson process with rate $\lambda (> 0)$ if it has following properties:

- $N(0) = 0$.
- $N(t)$ has independent increments (the numbers of occurrences counted in disjoint intervals are independent with each other)
- $N(t + \tau) - N(t) \sim Poi(\lambda\tau); \forall \tau, t > 0$.



Table 1. Characteristic functions of software system

$a(t)$	Total number of faults
$b(t)$	Fault detection rate
$m(t)$	Expected number of fault detected by time t (mean value function)
$\lambda(t)$	Failure intensity

Definition 3. Non-homogeneous Poisson process

Non-homogeneous Poisson process is a Poisson process with rate parameter $\lambda(t)$ such that the rate parameter of the process is a function of time.

The variance of the NHPP can be obtained as follows:

$$\text{Var}[N(t)] = \int_0^t \lambda(\tau) d\tau \quad (1)$$

And the auto-correlation function is given by

$$\begin{aligned} \text{Cor}(\tau) &= E[N(t)]E[N(t+s) - N(t)] + E[N^2(t)] \\ &= \int_0^t \lambda(\tau) d\tau \int_0^{t+\tau} \lambda(\tau) d\tau + \int_0^t \lambda(\tau) d\tau \end{aligned} \quad (2)$$

2.2 General NHPP software reliability model calculation

Let's use some function to describe characteristics of system in Table 1. By time t , the system has $a(t)$ faults and $m(t)$ faults have been detected so it has $a(t) - m(t)$ remaining faults. With detection rate is $b(t)$, the equation that describes relationship among the number of faults detected in period Δt , the total remaining faults of system and the fault detection rate is:

$$m(t + \Delta t) - m(t) = b(t)[a(t) - m(t)]\Delta t + o(\Delta t) \quad (3)$$

where $o(\Delta t)$ is infinitesimal value with Δt : $\lim_{\Delta t \rightarrow 0} \frac{o(\Delta t)}{\Delta t} = 0$. Let $\Delta t \rightarrow 0$:

$$\frac{\partial}{\partial t} m(t) = b(t)[a(t) - m(t)] \quad (4)$$

If t_0 is the starting time of testing process, with initial conditions $m(t_0) = m_0$ and $\lim_{t \rightarrow \infty} m(t) = a(t)$, Pham shows that general solution of (3) is [1]:

$$m(t) = e^{-B(t)} \left[m_0 + \int_{t_0}^t a(\tau) b(\tau) e^{B(\tau)} d\tau \right] \quad (5)$$

where

$$B(t) = \int_{t_0}^t b(s) ds \quad (6)$$

The reliability $R(t)$, defined as the probability that there are no failures in the time interval $(0, t)$, is given by:

$$R(t) = \Pr\{N(t) = 0\} = e^{-m(t)} \quad (7)$$

In general, the reliability $R(x|t)$, the probability that there are no failures in the interval $(t, t+x)$, is given by

$$R(x|t) = \Pr\{N(t+x) - N(t) = 0\} = e^{-[m(t+x) - m(t)]} \quad (8)$$

And its density is given by

$$f(x) = \lambda(t+x) e^{-[m(t+x) - m(t)]} \quad (9)$$

Where

$$\lambda(t) = \frac{\partial}{\partial t} m(t) \quad (10)$$

2.3 Some existing NHPP SRGMs

2.3.1 Goel Okumoto. The general assumptions of the GO model are [2]:

- (1) All faults in a program are mutually independent from the failure detection point of view.
- (2) The number of failures detected at any time is proportional to the current number of faults in a program. This means that the probability of the failures for faults actually occurring, i.e., detected, is constant.
- (3) The isolated faults are removed prior to future test occasions.
- (4) Each time a software failure occurs, the software error that caused the failure is immediately removed, and no new errors are introduced.

Especially, the GO-model assumes that the failure process is modelled by an NHPP model with mean value function $m(t)$ given by

$$m(t) = a(1 - e^{-bt}), a > 0, b > 0 \quad (11)$$

Where a is the expected total number of faults that exist in the software before testing and b is the failure detection rate or the failure intensity of a fault. Note that for this model: $m(\infty) = a$ and $m(0) = 0$.

Since $m(\infty)$ is the expected number of faults which will eventually be detected, the parameter a is then the final number of faults that can be detected by the testing process. The quantity b which is a constant of proportionality, can be interpreted as the failure occurrence rate per fault. The intensity function $\lambda(t)$ defined as the derivative of $m(t)$ is then

$$\lambda(t) = \frac{\partial}{\partial t} m(t) = abe^{-bt} \quad (12)$$

2.3.2 Yamada Delayed S-shaped. Let discuss about a stochastic model for a software error detection process based on NHPP in which the growth curve of the number of detected software errors for the observed failure data is S-shaped, called delayed S-shaped NHPP model [3].

The Yamada Delayed S-Shaped model is a modification of the non-homogeneous Poisson process to obtain an S-shaped curve for the cumulative number of failures detected such that the failure rate initially increases and later (exponentially) decays. It can be thought of as a generalized exponential model with failure rate first increasing and then decreasing. The software error detection process described by such an S-shaped curve can be regarded as a learning process because the testers' skills will gradually improve as time progresses.

The delayed S-shape model is based on the following assumptions:

- (1) All faults in a program are mutually independent from the failure detection point of view.
- (2) The probability of failure detection at any time is proportional to the current number of faults in a software.
- (3) The proportionality of failure detection is constant.
- (4) The initial error content of the software is a random variable.
- (5) A software system is subject to failures at random times caused by errors present in the system.
- (6) The time between $(i-1)^{\text{th}}$ and i^{th} failures depends on the time to the $(i-1)^{\text{th}}$ failure.
- (7) Each time a failure occurs, the error that caused the failure is immediately removed and no other errors are introduced.

Assume

$$b(t) = \frac{b^2 t}{bt + 1} \quad (13)$$



Where b is the error detection rate per error in the steady-state. The mean value function is given by

$$m(t) = a[1 - (1 + bt)e^{-bt}], a > 0, b > 0 \quad (14)$$

And the corresponding failure intensity function is

$$\lambda(t) = ab^2te^{-bt} \quad (15)$$

2.3.3 Musa-Okumoto. Musa-Okumoto model has been observed that the reduction in failure rate resulting from repair action following early failures are often greater because they tend to the most frequently occurring once [4], and this property has been incorporated in the model. The mean value function and intensity function of the model given as

$$m(t) = a \ln(1 + bt), a > 0, b > 0 \quad (16)$$

And

$$\lambda(t) = \frac{ab}{1 + bt} \quad (17)$$

where a is the expected total number of faults to be eventually detected and b is the fault detection rate.

There are some SRGMs introduced with different applications. Because of the limitations in theoretical and numerical computing, each model focuses on some fields of real assumptions. So practitioner need to analyse the application of each one in specific environment and evaluate them by experimental results to support other practitioners.

3. RANKING METHODOLOGY

As mentioned earlier, there are some NHPP SRGMs that have been introduced. Choosing one of them to apply in a specific problem is the subject of this section [6].

3.1 Comparison criteria

There are several approaches to evaluate the fitting between calculated values of SRGMs and a real data set. A set of used comparison criteria will be discussed as follows:

- (1) The mean square error (MSE) is the average of the squares of the difference between the estimated value and the observed total number of detected software errors:

$$MSE = \frac{\sum_{i=1}^k [m(t_i) - m_i]^2}{k - p} \quad (18)$$

- (2) The mean absolute error (MAE) is the average value of the absolute errors:

$$MAE = \frac{\sum_{i=1}^k |m(t_i) - m_i|}{k - p} \quad (19)$$

- (3) The mean difference error (MDE) is similar to MAE, but not using constant p :

$$MDE = \frac{\sum_{i=1}^k |m(t_i) - m_i|}{k} \quad (20)$$

- (4) The accuracy of estimation (AE) is the difference between the estimated number of all software errors a with the total number of all observed software errors M_a :

$$AE = \left| \frac{M_a - a}{M_a} \right| \quad (21)$$

- (5) The noise reflects the change of failure intensity:

$$Noise = \sum_{i=1}^k \left| \frac{\lambda(t_i) - \lambda(t_{i-1})}{\lambda(t_{i-1})} \right| \quad (22)$$

- (6) The predictive-ratio risk (PRR) is the total sum of the relative comparison between the estimated value and the observed total number of detected software errors:

$$PRR = \sum_{i=1}^k \frac{m(t_i) - m_i}{m(t_i)} \quad (23)$$

- (7) The sum of squared errors (SSE) is the total sum of the difference between the estimated value and the observed total number of detected software errors:

$$SSE = \sum_{i=1}^k [m(t_i) - m_i]^2 \quad (24)$$

- (8) The sum of difference (SD) is the total sum of the difference between the estimated value $s(t_i)$ and the actual occurrence time of each software error s_i :

$$SD = \sum_{i=1}^k |s(t_i) - s_i| \quad (25)$$

Note: k is the total number of records of the data set, and p is the number of parameters of each SRGM.

3.2 Ranking procedure

To evaluate the quality of every SRGMs, some comparison criteria have been introduced. Each of them has its advantages and limitations. Therefore, it should be a mathematics method to calculate a generalised measure that contains the contribution of every criteria. This sub-section will present this ranking procedure with four-step scenario.

3.2.1 Calculate measurements of every criterion of each SRGM. Let us consider n SRGMs with m criteria. Therefore, those measurements will be stored in $(n + 2) \times m$ matrix. In this criteria value matrix, each element a_{ij} shows the value of j^{th} criterion of i^{th} model. This matrix can be given as follows:

Criteria value matrix =

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \\ (Amin)_1 & (Amin)_2 & \cdots & (Amin)_m \\ (Amax)_1 & (Amax)_2 & \cdots & (Amax)_m \end{bmatrix} \quad (26)$$

Where:

- $(Amax)_j$ = maximum value of j^{th} criterion
- $(Amin)_j$ = minimum value of j^{th} criterion
- a_{ij} = value of j^{th} criteria of i^{th} model

3.2.2 Calculate the weight of each measurement. Contribution of each criterion is different. So those measurements have to be multiplied by a weight value. Lets consider the criterion of rating with two cases as follows:

- (1) Case 1: The smaller value of the criterion, the better quality of the SRGM:

$$Criterion \text{ rating} = \frac{MaC - VC}{MaC - MiC} \quad (27)$$

- (2) Case 2: The bigger value of the criterion, the better quality of the SRGM:

$$Criterion \text{ rating} = \frac{VC - MiC}{MaC - MiC} \quad (28)$$

Where

- MaC = the Maximum value of this Criterion



- MiC = the Minimum value of this Criterion
- VC = the Value of the Criterion

Let us consider X_{ij} represents the rating of j^{th} criterion of i^{th} SRGM. So have two equations from those cases:

$$X_{ij} = \frac{(Amax)_j - a_{ij}}{(Amax)_j - (Amin)_j} \quad (29)$$

Or

$$X_{ij} = \frac{a_{ij} - (Amin)_j}{(Amax)_j - (Amin)_j} \quad (30)$$

where $i, j \in \{1, 2, \dots, n\}$. This weight of criterion value can be calculated by:

$$W_{ij} = 1 - X_{ij} \quad (31)$$

Then the weight matrix can be represented as:

$$\text{Weight matrix} = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1m} \\ W_{21} & W_{22} & \dots & W_{2m} \\ \vdots & \vdots & \dots & \vdots \\ W_{n1} & W_{n2} & \dots & W_{nm} \end{bmatrix} \quad (32)$$

3.2.3 Calculate weighted value of each measurement. Weighted criteria value is calculated by multiplying the weight of each criterion with the criteria value i.e.

$$A_{ij} = W_{ij} \times a_{ij} \quad (33)$$

with $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, m\}$. Then:

Weighted criteria value matrix =

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & A_{22} & \dots & A_{2m} \\ \vdots & \vdots & \dots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nm} \end{bmatrix} \quad (34)$$

3.2.4 Calculate generalised measure of each SRGM. The generalised measure of each SRGM is given as follows:

$$Z_j = \frac{\sum_{i=1}^m A_{ij}}{\sum_{i=1}^m W_{ij}} \quad (35)$$

where $i \in \{1, 2, \dots, n\}$. This measure is based on all of 8 criteria that are introduced in previous sub-section. By using this calculation, a smaller measurement represents a better fitting of SRGM.

4. EXPERIMENTAL RESULT

Let see the computation as follows.

4.1 Implemented data sets

4.1.1 NTDS data set. The software data set listed below was extracted from information about failures in the development of software for the real-time multi-computer complex of the US Naval Fleet Computer Programming Center of the US Naval Tactical Data Systems (NTDS) [2]. The software consists of 38 different project modules. The time horizon is divided into four phases: production phase, test phase, user phase, and subsequent test phase. The 26 software failures were found during the production phase, five during the test phase and; the last failure was found on 4 January 1971. One failure was observed during the user phase, in September 1971, and two failures during the test phase in 1971. Data set is shown in Table 2 with:

- TBE is Time between errors: x_k (days)
- CT is Cumulative time $S_n = \sum x_k$ (days)

Table 2. Software failure data set (DS1)

Num	TBE	CT	Num	TBE	CT
Production (checkout) phase					
1	9	9	14	9	87
2	12	21	15	4	91
3	11	32	16	1	92
4	4	36	17	3	95
5	7	43	18	3	98
6	2	45	19	6	104
7	5	50	20	1	105
8	8	58	21	11	116
9	5	63	22	33	149
10	7	70	23	7	156
11	1	71	24	91	247
12	6	77	25	2	249
13	1	78	26	1	250
Test phase					
27	87	337	30	9	405
28	47	384	31	135	540
29	12	396			
User phase					
32	258	798			
Re-test phase					
33	16	814	34	35	849

Table 3. Software failure data set (DS2)

Num	TBE	CT	Num	TBE	CT
1	10	10	9	22	125
2	9	19	10	25	150
3	13	32	11	19	169
4	11	43	12	30	199
5	15	58	13	32	231
6	12	70	14	25	256
7	18	88	15	40	296
8	15	103			

Table 4. Parameter estimation of SGRMs for DS1

Model Name	MLE of Parameter	
Goel Okumoto	$a = 29.428768$	$b = 0.0074023404$
Goel Okumoto ³	$a = 33.993507$	$b = 0.0057901596$
Musa Okumoto	$a = 15.593125$	$b = 0.0137961535$
Musa Okumoto ³	$a = 23.397352$	$b = 0.008152455$
Delayed S-shaped	$a = 27.408987$	$b = 0.018325329$
Delayed S-shaped ³	$a = 27.49421$	$b = 0.018570999$

4.1.2 IBM data set of Ohba. Data in Table 3 is about testing an on-line data entry software package of IBM [7].

4.2 Ranking SRGMs with data set 1

4.2.1 Parameter estimation of DS1. Let have to use those criteria to rank 3 different SRGMs:

- (1) Goel-Okumoto model
- (2) Goel-Okumoto model³
- (3) Yamada Delayed S-shaped Model
- (4) Yamada Delayed S-shaped Model³
- (5) Musa-Okumoto Model
- (6) Musa-Okumoto Model³

Model with super-script "3" is the modification of each model respectively by an additional assumption: instead of using n units as input, the modification models only use $n - 1$. By using MLE method, the estimated and optimal values of parameters are given in Table 4



Table 5. Weighted value of criteria of DS1

Weight value	Goel	Goel	Musa	Musa	Delayed	Delayed
value	Okumoto	Okumoto ³	Okumoto	Okumoto ³	S-shaped	S-shaped ³
MSE	6.2	5.75	12.04	8.917	4.88	4.667
MAE	0.6	0.833	1.96	1.5	1.68	1.583
MDE	1.963	0.704	2.556	1.074	1.630	0.481
AE	0.090	0.259	0.422	0.133	0.015	0.018
Noise	2.286	1.372	1.615	1.046	3.173	2.439
PRR	-2	0	0	0	1	1
SSE	155	138	301	214	122	112
SD	513.339	473.439	844.829	558.714	336.287	309.576

4.2.2 *Weight value of criteria of DS1.* Matrix of weight value of criteria of DS1 given in Table 5.

4.2.3 *Model permanent value and ranking of DS1.* With the values of criteria, a *permanent value* is calculated to rank those models in Table 6.

4.3 Different ranking with different data sets

From the comparison of rankings of the six SRGMs based on the values of all criteria, it is shown that the ranking of SRGMs varies with respect to the selection of criteria. Lets change from DS1 to DS2. By using weighted criteria methodology and base on the above eight criteria, we carry out a ranking action for four SRGMs that is given in Table 7. Those results show that the ranks of four models differ from the rank which is concluded using DS. No single model is the best suitable one for all comparison criteria. In order to avoid this problem, it is proposed to apply weighted criteria methodology to analyse the performance and rank the SRGMs based on all these eight criteria taken collectively.

4.4 Ranking predictability of SRGMs

To calculate the accuracy of this application, 9 different data sets and 3 modified models will be applied to generate s_n value. The accuracy testing is exhibited as following steps:

- (1) Calculate ranking of modified models by using $n - 1$ units in dataset.
- (2) Calculate s_n by using calculated parameters from previous step.
- (3) Calculate the absolute deviation (AD) between real s_n and calculated s_n .

$$AD = |s_n - \hat{s}_n| \quad (36)$$

- (4) Calculate new rank based on the value of absolute deviation of each model.
- (5) Compare the former and the later ranks.

And the results are in Table 8. Those results show that there are no relationship between the fit of calculated occurrence failure times and the fit of a predicted time of the next failure. Not all of the models rated as 1st rank can predict the most precise value, only 5 out of 9 cases are operated correctly. For example, model Goel-Okumoto has established ranking in every data set. Model Musa-Okumoto has $\frac{5}{9}$ different ranking value in two areas, and Delayed S-shaped has $\frac{4}{9}$.

5. CONCLUSIONS AND FUTURE WORKS

Our work focuses on the implementation of some SRGMs with several real data sets to analyse and rank them. From experimental results, we summary that each SRGM has its own advantages in specific conditions. More specifically, a SRGM can be better in this case but do not supply a good value in another case. The

second part of our work introduces the predictability of SRGM and evaluate this prediction among SRGMs.

The weighted criteria method uses a relatively simple mathematical formulation. To determine the priority of criteria, we need to apply this weighted matrix method on many data sets. The method has the flexibility to choose the criteria to obtain the final decision. However, we need to conduct more improvements to increase the accuracy of this calculation. We have proposal for this problem: the assumptions of each model are different, so that the relationship between system faults is not simultaneous under different models' vision. If we can get more information about the system, such as the architecture and components, it is more available to classify the data sets, therefore we can apply the right group of models to calculate system faults.

6. REFERENCES

- [1] H. Pham, *System software reliability*. Springer, 2006.
- [2] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *Reliability, IEEE Transactions on*, vol. 28, no. 3, pp. 206–211, 1979.
- [3] S. Yamada, M. Ohba, and S. Osaki, "S-shaped software reliability growth models and their applications," *Reliability, IEEE Transactions on*, vol. 33, no. 4, pp. 289–292, 1984.
- [4] J. D. Musa and K. Okumoto, "A logarithmic poisson execution time model for software reliability measurement," in *Proceedings of the 7th international conference on Software engineering*, pp. 230–238, IEEE Press, 1984.
- [5] M. Lefebvre, *Applied probability and statistics*. Springer, 2006.
- [6] M. Anjum, M. A. Haque, N. Ahmad, *et al.*, "Analysis and ranking of software reliability models based on weighted criteria value," *International Journal of Information Technology and Computer Science*, vol. 2, no. 1, 2013.
- [7] M. Ohba, "Software reliability analysis models," *IBM Journal of research and Development*, vol. 28, no. 4, pp. 428–443, 1984.



Table 6. Permanent values & ranking of DS1

Model	Sum of Weight	Sum of Weight Value	Permanent Value	Rank
Goel Okumoto	3.5806112	398.3093	111.2406	4
Goel Okumoto ³	3.5611727	332.605	93.3976	3
Musa Okumoto	6.6277733	1163.468	175.5443	6
Musa Okumoto ³	4.3656235	530.2611	121.4629	5
Delayed S-shaped	4.501992	191.7318	42.58821	1
Delayed S-shaped ³	3.7265868	160.9457	43.1885	2

Table 7. Permanent values & ranking of DS2

Model Name	Sum of Weight	Sum of Weight Value	Permanent Value	Rank
Goel Okumoto	3.2196808	30.03869	9.329711	2
Goel Okumoto ³	2.7756686	23.69441	8.536469	1
Musa Okumoto	2.4449615	26.675	10.91019	4
Musa Okumoto ³	1.9075694	18.99226	9.956264	3
Delayed S-shaped	3.532439	67.42664	19.08784	6
Delayed S-shaped ³	2.7938755	50.00858	17.89936	5

Table 8. Ranking predictability of SRGMs

Data set	Factor	Goel Okumoto	Musa Okumoto	Delayed S-shaped
015_Ohba	AD	10.514099	11.996002	31.368652
	Former Rank	1	2	3
	Later Rank	1	2	3
021_Ehrlich	AD	65.85895	78.40881	174.82648
	Former Rank	1	2	3
	Later Rank	1	2	3
027_Naval	AD	63.91815	70.7272	16.135468
	Former Rank	2	3	1
	Later Rank	2	3	1
032_Musa6	AD	3.0061836	3.0768776	6.2299232
	Former Rank	1	2	3
	Later Rank	1	2	3
036_Musa14C	AD	8.792725	5.8891907	22.966995
	Former Rank	2	3	1
	Later Rank	2	1	3
037_Musa17	AD	20.427448	20.511261	18.97799
	Former Rank	1	3	2
	Later Rank	1	2	3
040_Musa4	AD	39.090942	22.717957	203.45105
	Former Rank	1	2	3
	Later Rank	2	1	3
041_Musa27	AD	11.553833	11.781399	6.748337
	Former Rank	2	1	3
	Later Rank	2	3	1
043_Musa2	AD	8.81332	8.890347	5.233288
	Former Rank	2	1	3
	Later Rank	2	3	1