

A Comparative Study of Failure Data for Software Reliability Estimation

¹S. Charles Ilayaraja, ²Dr. M. Ganaga Durga

¹Sethu Institute of Technology, Virudhunagar, India

²Government Arts College for Women, Sivagangai, India

Abstract

Software reliability is one of the attributes of quality and its measurement is supported by Software Reliability Growth Models. Among many of the proposed SRGMs, some of the models have been widely used and few of them obsolete. The successful SRGMs are characterized by the accuracy of reliability estimation. In such models, the reliability estimation depends on the quality of failure data and its accuracy increases as the length of failure observation increases. In this paper, we have studied the role of failure data set in the reliability estimation by employing three different data sets on unified SRGM that has no distinction between failure observation and fault removal process. The selected SRGM is incorporated with two distribution functions one is exponential and another one is 2-staged erlang distribution. For easy implementation, we assume that the debugging process is perfect. That is the probability of error correction is one and no errors have been introduced during the debugging process. Three different cases of failure data are considered for our study such as database application software, web server, the interface of an operating system. The unknown parameters of each case are estimated by using SMERFS tool and the goodness-of-fit analysis is done in MATLAB environment.

Keywords

Fault Count Data, Software Reliability Growth Model, Failure Intensity Function, Software Reliability Function, Non-Homogeneous Poisson Process

Acronym

TNOFR Total Number of Faults Remaining

TNOF	Total Number of Faults
RF	Reliability Factor
SSE	Sum of Squared Error
RMSE	Root Mean Square Error
SMERFS	Statistical Modeling and Estimation of Reliability Functions

Notation

$m(t)$	mean value function
$\lambda(t)$	Failure Intensity function
$\lambda(t+\delta x)$	Failure Intensity function for the next δx time units
$m(t+\delta x)$	mean value function for the next δx time units
$R(t \delta x)$	Reliability function for the next δx time units
$b(t)$	Fault detection rate
a	initial number of faults
b	proportionality constant
p	probability of error generation
α	error generation rate
t	failure time / failure interval
$\sum_{i=1}^n f_i$	total number of faults observed at n time intervals
t_n	n^{th} time interval
$\bar{f}$	mean of the observed faults
f_i	faults observed at i^{th} time

$m(t_i)$ mean value function at i^{th} time

I. Introduction

Software failures have been formally classified with respect to their criticality to the system operation and maintenance as expected failures, configuration failures and critical failures [8]. Expected failures are caused by frequent system start, frequent migration, system update and unplanned code re-engineering. These failures could be managed by using appropriate documents and schedules. The second class of failures is configuration failures which are caused by incomplete installation, incompatible environment and version conflicts within the system. This can be addressed by having proper configuration control and management. The third class of failures is critical failures that need consistent solution. Since the causes of such failures are unexpected, there is need for a model based mechanism to forecast

- System crashes
- Hardware failures
- Failures in application associations and linkages
- Violations in communication and interfaces

In order to address the above issues, many SRGMs have been proposed in the literature. Reliability models predict the intensity of failure and the cumulative number of faults at the specified time instant. With respect to time instant whether calendar time or CPU execution time, software failure data are classified into Time Between Failure Data and Fault Count Data. The former class of failure data is described by the test interval and the later class of failure data is described by the fault count at the specified time instant. The SRGMs for each class of failure data differ by their initial assumptions [5].

Poisson process models are used to describe the occurrences of random failures during the testing phase [13]. The failure counting process is characterized by the failure rate represented by the parameter say λ . In practical, software failure rate is non-linear, the failure counting process is modeled by Non-Homogeneous Poisson Process given in [1]

$$P = \frac{\lambda^k}{k!} e^{-\lambda}, \quad k = 0, 1, 2, \quad (1)$$

Two versions of unified SRGM models have been proposed and we select generalized NHPP model that treats both failure observation and fault removal process as same function. The first version of unified SRGM in given in [1] as follows

$$m(t) = \frac{a}{1-\alpha} [1 - (1 - F(t))^{p(1-\alpha)}] \quad (2)$$

In the above model, $F(t)$ is the generalized representation of failure observation or fault removal process and there is no distinction between failure observation process and fault removal process. We employ two failure distribution functions, the exponential and two-staged erlang distribution into the $F(t)$ of (2) to derive NHPP model and delayed S-shaped model. For the derivation purpose, we have the following assumptions on (2)

- The Failure process follows NHPP
- Each fault contributes failure at same severity level
- No errors introduced during the failure observation. i.e.) $\alpha=0$
- The observed faults are corrected with the probability of one. i.e.) $p=1$
- Each time a failure is observed, an immediate debugging effort takes place without any delay.

For a given constant failure rate, the exponential distribution function with respect to the specified time instant is given by

$$F(t) = 1 - e^{-bt} \quad (3)$$

Now (3) is incorporated in (2) then we have

$$m(t) = \frac{a}{1-\alpha} [1 - e^{-bp(1-\alpha)t}] \quad (4)$$

Similarly, the two-staged erlang distribution is given by

$$F(t) = 1 - (1 + bt)e^{-bt} \quad (5)$$

which is incorporated in (2) we have

$$m(t) = \frac{a}{1-\alpha} [1 - ((1 + bt)e^{-bt})^{p(1-\alpha)}] \quad (6)$$

By using the model assumptions such as $\alpha=0$ and $p=1$ then the (4) and (6) becomes

$$m(t) = a(1 - e^{-bt}) \quad (7)$$

$$m(t) = a[1 - ((1 + bt)e^{-bt})] \quad (8)$$

Here (7) represents NHPP SRGM model and (8) represents delayed S-shaped model. Table 1 summarizes the characteristics of both NHPP and Delayed S-shaped SRGM models [1].

Table 1: Summary of selected two SRGMs

Model Name	Mean Value Function $m(t)$	Failure Intensity Function $\lambda(t)$
NHPP Model	$a(1-e^{-bt})$ $a>0, b>0$	abe^{-bt}
S-shaped Model	$a[1-(1+bt)e^{-bt}]$ $a>0, b>0$	ab^2te^{-bt}

A. Parameter Estimation

For both models, the proportionality constant is estimated by using specific tool SMERFS 3.0 and the other parameter is computed using MLE functions. The first parameter represents initial number of faults at the start of failure observation and the second proportionality constant parameter represents failure rate that occurs during the failure observation. For each data set and each model, the value of proportionality constant is obtained from SMERFS output. By using the value of proportionality constant that is obtained for each SRGM model and for each Data Set, the initial faults can be computed for each model. For NHPP models the value is obtained by using (9) and for delayed S-shaped model

the value is obtained by using (10).

$$a = \frac{\sum_{i=1}^n f_i}{(1 - e^{-bt_n})} \quad (9)$$

$$a = \frac{\sum_{i=1}^n f_i}{1 - ((1 + bt_n)e^{-bt_n})} \quad (10)$$

After obtaining the values of unknown parameters of the two selected SRGM models, their $\lambda(t)$, $m(t)$ are computed. The failure intensity function describes the faults residuals and the mean value function of each model describes the expected number of faults at each time instant within the length of failure observation. By using $m(t)$ of each model, the reliability function $R(t | \delta x)$ is computed as follows

$$R(t | \delta x) = e^{-(m(t+\delta x)-m(t))} \quad (11)$$

II. Related Work

A set of software metrics have been proposed to account for information gained from a user's point view regarding the severity of the observed failures. By adapting such metric set, an extended Modified Adaptive Strategy (MAT) has been proposed for better estimation of reliability [14]. A generalized SRGM model has been proposed based on the classification of the faults with respect to their removal complexities. This model addresses the problems in the imperfect debugging and error generation rate and supports for realistic estimation [11].

A scheme has been proposed for constructing SRGM based on NHPP where testing effort is described as Generalized Exponential Distribution (GED) with the assumption that the error detection rate is proportional to the current fault content [12]. Software Testing strategies have been analyzed with respect to the attained failure size to reveal the impact of software testing effort and the failure data volume [9].

The importance of residuals and the total number of expected faults which are taken into account for the measurement of software reliability is addressed in [6,8]. Both fault detection and removal process are integrated and a new SRGM has been modeled based on the integration in [3,7]. A quantitative approach for the measurement of reliability on open source software systems has been proposed and analyzed to address the issues on the optimal release policies. The measurement of reliability has been studied and implemented on two open source software systems MySQL and Linux Kernel [4,15-16].

A Unified approach has been incorporated for fault modeling and two versions of unified SRGM model have been proposed by incorporating both imperfect debugging and error generation. The first version of unified SRGM has no distinction between failure observation and fault removal and the second version models both are different. Both versions of unified SRGMs have good applicability by supporting many of the failure distributions [1-2].

III. Data Analyses

Three data sets have been taken for the comparative analysis of software reliability estimation. DS-I represents the failure data obtained from PL/I database application software where 328 faults have been reported in 19 weeks [3]. The remaining two data sets DS-II and DS-III have been obtained from famous OSS projects

of their first release. DS-II represents failure data of GNOME project of its first version 2.0 where 74 faults have been reported in 24 weeks [4]. DS-III represents failure data of Apache project of its first version 2.0.35 where 85 faults have been reported in 35 days [4].

A. Model Validation

Both models can be validated on the given three data sets using curve fit analysis. The following criteria have been taken in account to understand the goodness of fit for three data sets to be employed with both NHPP model and Yamada S-shaped model.

1. SSE (Sum of Squared Error)

It is the sum of squared difference between the estimated curve and the actual data. The smaller value of SSE indicates better fit. It is defined in [1] as

$$SSE = \sum_{i=1}^n (f_i - m(t_i))^2 \quad (12)$$

2. R² (R-Square)

It is the ratio of sum of squared in the regression about the mean. The R² value can take any value between 0 and 1 where the value which is closer to 1 has better fit. It is defined in [1] as

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(t_i) - \bar{f})^2}{\sum_{i=1}^n (f_i - \bar{f})^2} \quad (13)$$

3. RMSE (Root Mean Square Error)

It is the measure of closeness with which the model prediction and the observation. The lower value indicates better fit. It is defined in [1] as

$$Bias = \frac{\sum_{i=1}^n (m(t_i) - f_i)}{n} \quad (14)$$

$$RMSE = \sqrt{Bias^2 + Variance^2} \quad (15)$$

$$Var = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - m(t_i) - Bias)^2} \quad (16)$$

For DS-I, S-shaped model (M-II) has better fit than NHPP (M-I) model. For DS-II and DS-III, a negative R² was generated in NHPP model which shows non-applicability of R² and remaining criteria of goodness of fit are same in both NHPP and S-shaped models. To avoid negative R² value, data set has to be normalized to some measurement unit. Table 2 to 4 shows the goodness of fit analysis of three data sets on two SRGM models M-I and M-II. Table 2: Curve fit analysis for DS-I

	DS - I				
	A	B	SSE	R ²	RMSE
M-I	516.79	0.053	2503.492	0.00087	12.14
M-II	360.26	0.212	2502	0.00143	12.13

Table 3: Curve fit analysis for DS-II

	DS - II				
	A	B	SSE	R ²	RMSE
M-I	118.0999	0.053	142	NA	2.54
M-II	91.2064	0.182	142	NA	2.54

Table 4: Curve fit analysis for DS-III

	DS - III				
	A	B	SSE	R ²	RMSE
M-I	87.7248	0.053	133.5	NA	2.012
M-II	76.5093	0.15	133.5	2.22E-16	2.012

IV. Model Performance Analysis

The performance of both models on the given three data sets has been illustrated graphically by using MATLAB 9.0. Each data set is employed with both NHPP and delayed S-shaped model to generate m(t) and failure intensity plots. From each plot, we describe the performance as follows;

A. Data Set-I

1. m(t) is better for NHPP model than Yamada S-shaped model. At m(tn), the prediction plot of NHPP model is very close to the total number faults where as plot of S-shaped model deviated as in fig. 1(a)
2. λ(t) is constantly decreases as time increases for NHPP model where as failure intensity increases and then decreases than NHPP model because Yamada S-shaped model considers the delay between failure observation and fault removal as in fig. 1(b). The reduced failure intensity has good estimate of reliability.

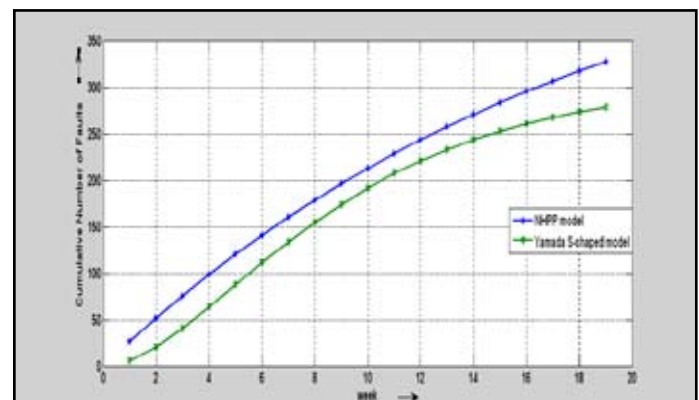


Fig. 1(a) Mean Value Function Plot for DS-I

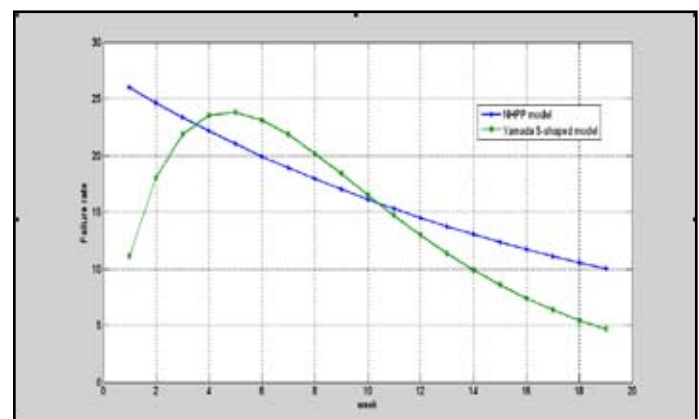


Fig. 1(b): Failure Intensity Plot for DS-I

B. Data Set-II

1. At the start of failure observation, $m(t)$ for both NHPP and Yamada S-shaped models is very close to each other and then considerably deviates at $m(t_n)$ as in fig. 2(a)
2. Alternately $\lambda(t)$ for both models have deviation and close to each other at $m(t_n)$ as in fig. 2(b)

C. Data Set-III

1. At $m(t_1)$, the prediction plot for both models are closer together, but have much deviation at $m(t_n)$ as graphically illustrated in fig. 3(a).
2. The failure intensity at both $m(t_1)$ and $m(t_n)$ are closer for each model and little deviation is present in between t_1 and t_n as in fig. 3(b)

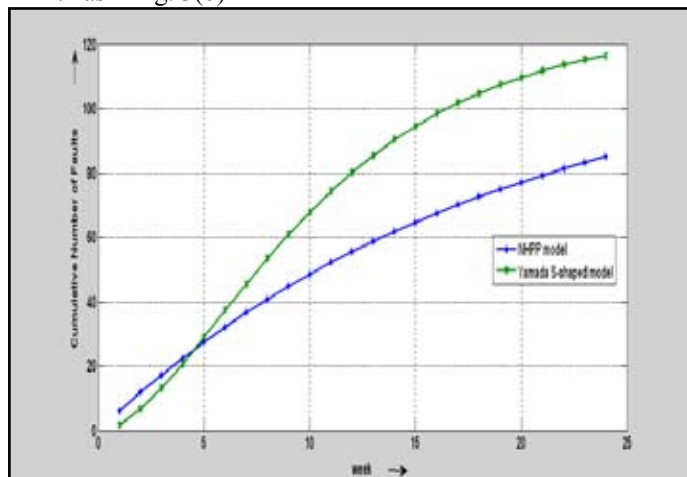


Fig. 2(a): Mean Value Function Plot for DS-II

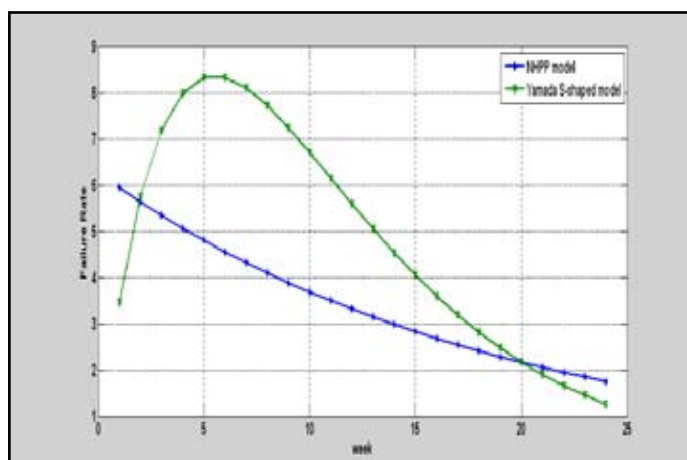


Fig. 2(b): Failure Intensity Plot for DS-II

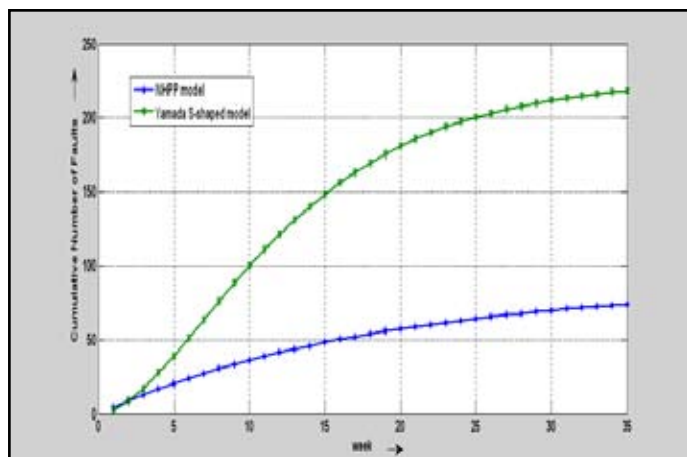


Fig. 3(a): Mean Value Function Plot for DS-III

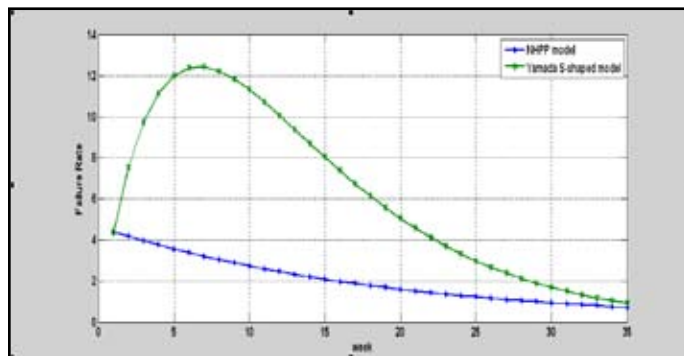


Fig. 3(b): Failure Intensity Plot for DS-III

From the above observations, Yamada S-shaped model is more suitable for practical software development environment than NHPP model. This is because one of the NHPP models assumptions is that an immediate effort is taken as soon as faults are identified which is impractical.

As Yamada delayed S-shaped model addresses this issue by considering such delay in its $m(t)$ and hence the prediction plot has considerable deviation than NHPP model. To have more understanding, the reliability function $R(t|\delta x)$ is computed for next day ($\delta x=0.1429$ week) in DS-I and DS-II and for next hour ($\delta x=0.04167$ day) in DS-III using MATLAB environment. Table 5 & 6 illustrated this performance of both models on three data sets.

Table 5: Reliability Estimation Using NHPP

	NHPP model				
	$\lambda(t)$	$m(t)$	$\lambda(t+\delta x)$	$m(t+\delta x)$	$R(t \delta x)$
DS – I $t=19$ $\delta x=0.143$	10.01	328	9.93	329.42	0.2407
DS – II $t=24$ $\delta x=0.143$	1.75	84.99	2.27	85.25	0.7791
DS – III $t=35$ $\delta x=0.042$	0.73	73.99	0.71	74.29	0.741

Table 6: Reliability Estimation Using Delayed S-Shaped Model

	Yamada S-shaped model				
	$\lambda(t)$	$m(t)$	$\lambda(t+\delta x)$	$m(t+\delta x)$	$R(t \delta x)$
DS – I $t=19$ $\delta x=0.143$	5.48	327.99	5.36	328.77	0.46
DS – II $t=24$ $\delta x=0.143$	0.92	85	0.90	85.13	0.88
DS – III $t=35$ $\delta x=0.042$	0.32	74	0.30	74.13	0.88

Alternately, the reliability is computed numerically as cited in [10]

$$RF = 1 - \left[\frac{TNOFR}{TNOF} \right] \quad (17)$$

Here, TNOFR represents the number residuals and TNOF represents total number of expected faults. The values for both TNOFR and TNOF are obtained by using SMERFS 3.0. Irrespective of the model application (NHPP or S-shaped) and test duration (Weeks/ Days), the reliability is computed numerically and we observe that the higher reliability is obtained for DS-III. Table 7 illustrates the results obtained from SMERFS tool.

Table 7: Reliability Estimation Using Fault Residuals

	Length	CF	TNOFR	TNOF	RF
DS-I	19	328	157.82	482.82	0.673128702
DS-II	24	85	30.962	112.96	0.725902975
DS-III	35	74	12.450	85.450	0.854300761

V. Conclusion

In this paper, three failure data sets have been selected and employed on two Poisson Process based SRGMs for software reliability estimation. First the three data sets have applied with SMERFS 3.0 and based on its output, the reliability is computed numerically. Then the same data sets have been employed on both NHPP and Delayed S-shaped models. The proportionality constant for each model has been obtained from SMERFS tool, and mean value functions for each model have been plotted using MATLAB environment. In both cases, we have studied that the quality of data set has considerable role in the software reliability estimation. Indeed, our study may motivate the researches to address the issues on failure data collection and characterization and reveals the possible risks in the initial parameter estimation for any of the SRGM models to be used for software reliability estimation.

References

- [1] P.K.Kapur, H.Pharm, Sameer Anand, Kalpana Yadav, "A Unified Approach for developing Software Reliability Growth Models in the presence of Imperfect Debugging and Error Generation", IEEE trans. on Reliability, Vol. 60, No. 1, 2011, pp. 331-340
- [2] Chin-Yu Huang, M.R. Lyu, Sy-yen Kuo, "A Unified Scheme of some NHPP models for Software Reliability Estimation", IEEE trans. on Software Engineering, Vol. 29, No. 3, 2003, pp. 261-269
- [3] Junj-Hua, Chin-Yu, "An Integration of Fault Detection and Correction Processes in Software Reliability Analysis", The Journal of Systems and Software, 2005, pp. 1312-1323
- [4] Xiang Li, Yan Fu Li, Min Xie, Szu Hui Ng, "Reliability Analysis and Optimal Version Updating for Open Source Software", Information and Software Technologies, 2011, pp. 929-936
- [5] Razeef Mohd, Mohsin Nazir, "Software Reliability Growth Models: Overview and Applications", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 9, 2012, pp. 1309-1320
- [6] Ajay Gupta, Digvijay Choudary, Suneet Saxena, "Software Reliability Estimation Using Yamada Delayed S Shaped Model Under Imperfect Debugging and Time Lag", International Journal of Computer Applications, Vol. 23, No. 7, 2011, pp. 49-52.
- [7] N.Ahmed, S.M.K Quadri, Razeef Mohd, "Comparison of Predictive Capability of Software Reliability Growth Models with Exponentiated Weibull Distribution", International Journal of Computer Applications, Vol. 15, No. 6, 2011, pp. 40-43.
- [8] Pankaj Nagar, Blessy Thankachan, "Application of Goel-Okumoto Model in Software Reliability Measurement", Spl. Issue: International Journal of Computer Applications, ICNICT-2012, pp. 1-3
- [9] Babu Zachariah, "Analysis of Software Testing Strategies Through Attained Failure Size", IEEE Trans. On Reliability, Vol. 61, No. 2, 2012, pp. 569-579
- [10] Swapna S.Gokhale, M.R.Lyu, K.S.Trivedi, "Incorporating Fault Debugging Activities into Software Reliability Models", IEEE Trans. On Reliability, Vol. 55, No. 2, 2006, pp. 281-292.
- [11] Omar Shatnawi, P.K.Kapur, "A Generalized Software Fault Classification Model", WSEAS Trans. On Computers, Vol. 7, No. 9, 2008, pp. 1375-1384
- [12] S.M.K. Quadri, N.Ahmed, Sheikh Umar Farooq, "Software Reliability Growth Modeling with Generalized Exponential Testing – Effort and Optimal Software Release Policy", Global Journal of Computer Science and Technology, Vol. 11, No. 2, 2011, pp. 26-42
- [13] Richard Lai, Mohit Garg, "A Detailed Study of Non-Homogeneous Poisson Process Software Reliability Models", Journal of Systems and Software, Vol. 7, No. 6, 2012, pp. 1296-1306
- [14] Hai Hu, Chang-Hai Jiang, Kai-Yuan Cai, Eric Wong, A.P.Mathur, "Enhancing Software Reliability Estimates using Modified Adaptive Testing", Information and Software Technology, 2012, pp. 288-300
- [15] Sanjeev Kumar Jha, Shivlok Singh, Amod Tiwari, "Estimation and Analysis of MYSQL Database Server Reliability using Beta and Generalized Gamma Distribution", Int. Journal of Engg. Research & Technology, Vol. 1, No. 6, 2012, pp. 1-11.
- [16] Sanjeev Kumar Jha, A.K.D.Dwivedi, Amod Tiwari, "Reliability Estimation and Analysis of Linux Kernel", Int. Journal of Computer Science Technology, Vol. 2, No. 2, 2011, pp. 45-51.



S.Charles Ilayaraja received his B.Sc., degree in Computer Science from Madurai Kamaraj University College, Tamil Nadu, India, in 1998, the M.Sc., degree in Computer Science from Madurai Kamaraj University College, Tamil Nadu, India, in 2000, and the M.Phil., degree in Computer Science from Madurai Kamaraj University, Tamil Nadu, India in 2006. Having 13 years teaching experience, he has life-time membership with ISTE, India since

2008. His research interests include Software Reliability, Software Design and Architectures, Software Re-engineering, Quality Function Deployment and Theoretical Computer Science. At present, he is working as Assistant Professor, with Department of Applied Sciences in Sethu Institute Technology, Tamil Nadu, India since 2007.



Dr.M.Ganaga Durga received her B.Sc., degree from Madurai Kamaraj University, Tamil Nadu, India, in 1992, the M.C.A. degree from Bharathiyar University, Tamil Nadu, India, in 1995, and the M.Phil. degree in Computer Science from Mother Teresa Women's University, Tamil Nadu, India in 2002. She received Ph.D. degree in Mobile Network Security from Mother Teresa Women's University, Tamil Nadu, India, in 2009. With 18 years of teaching

experience, she worked as Professor in KLN College of Engineering about 10 years. She was functional Head for one of the Madurai Anna University Research Centers. She published two international journals, two national journals, four books and six conference papers. She has been invited as reviewer for three reputed journals. Her research interests include Mobile Network Security, Wireless Sensor Networks, Software Engineering, Cloud & Pervasive Computing. At present, she is working as Assistant Professor, with P.G Department of Computer Science in Government Arts College for Women, Sivagangai, Tamil Nadu, India since 2011.