I.J. Information Technology and Computer Science, **2013, 02, 1-14**
Published Online January 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2013.02.01

Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value

Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad
Department of Computer Engineering, Aligarh Muslim University, U.P.-202002, India
anjum.zhcet@gmail.com, asrafb4u@gmail.com, nesar.ahmad@gmail.com

Abstract**—** Many software reliability growth models (SRGMs) have been analyzed for measuring the growth of software reliability. Selection of optimal SRGMs for use in a particular case has been an area of interest for researchers in the field of software reliability. All existing methodologies use same weight for each comparison criterion. But in reality, it is the fact that all the parameters do not have the same priority in reliability measurement. Keeping this point in mind, in this paper, a computational methodology based on weighted criteria is presented to the problem of performance analysis of various non-homogenous Poisson process (NHPP) models. It is relatively simple and requires less calculation. A set of twelve comparison criteria has been formulated and each criterion has been assigned different weight to rank the software reliability growth models proposed during the past 30 years. Case study results show that the weighted criteria value method offers a very promising technique in software reliability growth models comparison.

Index Terms**—** Srgms, Software Reliability, NHPP, Software Model Ranking

I. Introduction

The software development process becomes increasingly time-consuming and expensive due to the complexity of software systems. In the mean time, the need for the highly reliable software system is ever increasing. How to enhance the reliability of the software systems and reduce the cost to an acceptable level becomes the main focus of the software industry. Methods of applying reliability and cost models to the software development practice are highly desired. Early work in the field of software reliability focused around proposing new models. Over the past 30 years, many SRGMs have been proposed for estimation of reliability growth of products during software development process [1-5]. Each model could be shown to work well with a unique data set, but no model appeared to do well on all data sets. Many researchers like Musa et al. have shown that some families of models have, in general, certain characteristics that are considered better than others [6-9]; for example, the geometric family of models tends to have better predictive quality than other models. Research is going on for finding the techniques to choose the best model for an individual application among the existing models. Ideally we would like to be able to select, before starting, which model we should use. But in reality, it is almost an impossible task. Brocklehurst et al. suggest that the variable nature of software failures has made the model selection process very difficult [10-14]. Software failures are caused by hidden design flaws and not by the psychological sciences that will someday show us how to select the model beforehand [14-17]. Today we must evaluate different models, compare them, and choose the best. Goel and Okumoto published a paper describing a non homogeneous Poisson process model from the finite exponential class of models [18-25]. This was one of the first non homogeneous Poisson process models proposed. Goel and Okumoto validated this model by showing that it predicted well on a unique data set. Goel and others started describing processes for which each model would be tested to see how well the model fits the data and predicts the future events. The assertion was that different models predict well only on certain data sets. By comparing the predictive quality of different models, it is possible to select the best one for a given application. Abdel-Ghaly et al. compared the predictive quality of 10 models using five different methods of comparison [20-22]. They showed that different methods of model selection result in different models being chosen. Also some of their comparison methods were subjective and too much complex. Clearly a simple and objective method to select models is needed. Khoshgoftar and Woodcock proposed a method to select a reliability model among various alternatives using the log-likelihood function [22-25]. They apply the method to the failure logs of a project. The method selected an S-shaped model as the most appropriate one. Till up to now, there is no such method that takes the weight of comparison criteria in selecting the appropriate model. All existing methods consider each comparison criteria with equal priority. But in real scenario, different criteria have different impact in measuring the software reliability. So to remove this flaw, we have developed a method in which we have taken the weight of each criterion into account for overall ranking of the models for a given failure data. In weighted criteria method we take any number of comparison-criteria with different weight for any number of models. After calculating the overall weighted value of different criteria for each model, we rank the models for a given failure data. The structure of our paper is organized as follows. Section 2 reviews few existing SRGMs. Section 3 gives the idea about comparison criteria. Section 4 explains our new proposed scheme. Section 5 analyzes the scheme (using case-study) on two different data sets. Section 5 concludes the paper with some remarks.

II. Description of Different SRGMs

The SRGMs have been widely used in analyzing and estimating the reliability related metrics of software products in many applications, such as telecommunications as described by Zhang *et al.* and Jeske *et al.* We have to know some parameters before going into the details of different models. Let {X(t), t≥0 } denote a counting process representing the cumulative number of faults detected by the time t. An SRGM based on an NHPP with the mean value function (MVF), *m*(t) can be formulated as: $\mathrm{P}\{\mathrm{X}(\mathrm{t}) = \mathrm{n}\} = e^{-[m(\mathrm{t})]} * \frac{[m(\mathrm{t})]^n}{n!}$ where n = 0, 1, 2, 3….. and m(t) represents the expected cumulative number of faults detected by the time t. The MVF *m*(t) is non-decreasing with respect to testing time t under the bounded condition m(∞) = *a*, where *a* is the expected total number of faults to be eventually detected. Knowing its value can help us to determine whether the software is ready to be released to the customers and how much more testing resources are required. It can also provide an estimate of the number of failures that will eventually be encountered by the customers. Generally, we can get distinct NHPP models by using different non-decreasing mean value functions. The failure intensity function at testing time t is $\lambda(\mathrm{t}) = \frac{\mathrm{d}m(\mathrm{t})}{\mathrm{dt}} = m'(\mathrm{t})$.

The software reliability, i.e., the probability that no failures occur in (s, s+t) given that the last failure occurred at testing time s (s≥0, t>0), is $\mathrm{R}(\mathrm{t}|\mathrm{s}) = \exp[-(m(\mathrm{t}+\mathrm{s}) - m(\mathrm{t}))]$

The fault detection rate per fault at testing time t is given by $\mathrm{d}(\mathrm{t}) = \frac{m'(\mathrm{t})}{a-m(\mathrm{t})} = \frac{\lambda(\mathrm{t})}{a-m(\mathrm{t})}$

There are several existing model well-known NHPP models with different MVFs, as describe below.

2.1 Goel-Okumoto Model

This model, first proposed by Goel and Okumoto, is one of the most popular NHPP model in the field of software reliability modeling. It is also called the exponential NHPP model. Considering failure detection as a Non homogeneous Poisson process with an exponentially decaying rate function, the mean value function is hypothesized in this model as

$$m(t) = a(1 - ex\,p[-bt]), \qquad a > 0, b > 0$$

and the intensity function of this model is given as

$$\lambda(t) = ab * exp[-bt], \qquad a > 0, b > 0$$

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

2.2 Gompertz Growth Curve Model

Gompertz Growth Curve Model is used in the Fujitsu and Numazu work. Many Japanese computer manufacturers and software houses have adopted this model since it is one of the simplest S-shaped software reliability growth models. Its mean value function and intensity function are

$$m(t) = ak^{b^t}, \qquad a > 0, 0 < b < 0, 0 < k < 1,$$

$$\lambda(t) = abln(k)\,k^{\exp[-bt]}exp[-bt], \qquad a > 0, 0 < b < 0, 0 < k < 1$$

where *a* is the expected total number of faults to be eventually detected and *b* and *k* are parameters whose values are estimated using regression analysis.

2.3 Logistic Growth Curve Model

In general, software reliability tends to improve and it can be treated as a growth process during the testing phase. That is, the reliability growth occurs due to fixing faults. Therefore, under some conditions, the models developed to predict economic population growth could also be applied to predict software reliability growth. These models simply fit the cumulative number of detected faults at a given time with a function of known form. Logistic growth curve model is one of them and it has an S-shaped curve. Its mean value function and intensity function are

$$m(t) = \frac{a}{1 + k * exp[-bt]}, \qquad a > 0, b > 0, k > 0$$

$$\lambda(t) = \frac{abkexp\,[-bt]}{(1 + k * exp[-bt])^2}, \qquad a > 0, b > 0, k > 0$$

where and a is the expected total number of faults to be eventually detected and k and b are parameters which can be estimated by fitting the failure data. Similar to the analysis given for the Gompertz curve, the point of inflection is $t_{\inf}= \ln(k)/b$. If $k > 1$, then $t_{\inf} > 0$ and we have an S-shaped results; when $0 < k \leq 1$, then $t_{\inf} \leq 0$ and no S-shaped curve is present. Therefore, the logistic reliability growth curve is S-shaped when $k > 1$ and $b > 0$.

2.4 Generalized Goel NHPP Model

In order to describe the situation that software failure intensity increases slightly at the beginning and then begins to decrease, Goel proposed a simple generalization of the Goel-Okumoto model with an additional parameter c. The mean value function and intensity function are

$$m(t) = a(1 - ex\,p[-bt^c]), \qquad a > 0, b > 0, c > 0$$

$$\lambda(t) = abct^{c-1}\,exp[-bt^c]\,, \qquad a > 0, b > 0, c > 0$$

where a is the expected total number of faults to be eventually detected and b and c are parameters that reflect the quality of testing.

2.5 Yamada Delayed S-Shaped Model

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 mean value function and intensity function are

$$m(t) = a(1 - (1 + bt) * ex\,p[-bt]), \qquad a > 0, b > 0$$

$$\lambda(t) = ab^2 t * exp\,[-bt], \qquad a > 0, b > 0$$

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

2.6 Inflected S-Shaped Model

This model solves a technical problem in the Goel-Okumoto model. It was proposed by Ohba and its underlying concept is that the observed software reliability growth becomes S-shaped if faults in a program are mutually dependent, i.e., some faults are not detectable before some others are removed. The mean value function is

$$m(t) = a * \frac{1 - exp\,[-bt]}{1 + \Psi(r) * exp[-bt]}, \qquad \Psi(r) = \frac{1 - r}{r}, \qquad a > 0, b > 0, r > 0$$

The parameter r is the inflection rate that indicates the ratio of the number of detectable faults to the total number of faults in the software, a is the expected total number of faults to be eventually detected, b is the fault detection rate, and is the inflection factor. On taking $\Psi(r) = \beta$ then the inflection S-shaped model mean value function and intensity function are given as

$$m(t) = a * \frac{1 - exp\,[-bt]}{1 + \beta * exp[-bt]}, \qquad a > 0, b > 0, \beta > 0$$

$$\lambda(t) = \frac{abexp\,[-bt](1 + \beta t)}{(1 + \beta * exp\,[-bt])^2}, \quad a > 0, b > 0, \beta > 0$$

2.7 Modified Duane Model

Duane published a report that presented failure data of several systems during their developments in 1962 by analyzing the data. It was observed that the cumulative Mean-Time Between-Failures versus the cumulative operating time becomes close to a straight line if plotted on log-log paper. Later, a modified Duane model was proposed and it's hypothesized mean value function and its intensity function are given as

$$m(t) = a\left\{1-\left(\frac{b}{b+t}\right)^{c}\right\}, \qquad a>0, b>0, c>0$$

$$\lambda(t) = acb^{c}(b+t)^{-(1+c)}, \qquad a>0, b>0, c>0$$

Where a is the expected total number of faults to be eventually detected.

2.8 Two-Error-Type Model

Yamada and Osaki proposed a software reliability growth model with non homogeneous fault detection rate by assuming that the errors can be divided into type 1 and type 2 errors. They assume that type 1 errors are easy to detect and type 2 errors are difficult to detect. The mean value function is

$$m(t) = a * \sum_{i=0}^{2} p(i)\,(1-exp\,[b(i)\,t]), \; a>0, 0<b(2)<b(1)<1, p(1)+p(2)=1, \qquad 0<p(1)<1, \; 0<p(2)<1,$$

where a is the expected total number of faults to be eventually detected, $b(i)$ is the fault detection rate of type i error (i = 1, 2) and $p(i)$ is the percentage of type i error (i = 1,2)

2.9 Weibull-Type Testing-Effort Function Model

Yamada et al. [3] proposed a software reliability growth model incorporating the amount of test effort expended during the software testing phase. The mean value function is

$$m(t) = a(1-exp[-b\alpha(1-exp[-\beta t^{\gamma}])]), \qquad a, b, \alpha, \beta, \gamma > 0$$

where a is the expected total number of faults to be eventually detected, α is the total amount of test effort required by software testing, β is the scale parameter, γ is the shape parameter, and b is the fault detection rate.

2.10 Musa-Okumoto Model

Musa-Okumoto have 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, and this property has been incorporated in the model. The mean value function and intensity function of the model given as

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

$$\lambda(t) = \frac{ab}{(1+bt)}, \qquad a>0, b>0$$

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

2.11 Yamada Imperfect Debugging Model 1

In general it is considered to be unrealistic in software reliability modelling to assume that the faults detected by software testing are perfectly removed without introducing new faults. Yamada et.al proposed software reliability assessment models with imperfect debugging by assuming that new faults are sometimes introduced when the faults originally latent in the software system are corrected and removed during the testing phase. It is assumed that the fault detection rate is proportional to the sum of the numbers of faults remaining originally in the system and faults introduced by imperfect debugging. This model is described by a non homogeneous Poisson process. The mean value function and intensity are given as

$$m(t) = a * b * \left(\frac{exp\,[\alpha t] - exp[-bt]}{\alpha + b}\right), \qquad a>0, b>0, \alpha>0$$

$$\lambda(t) = a * b * \left(\frac{\alpha * exp\,[\alpha t] + b * exp\,[-bt]}{\alpha + b}\right), \qquad a>0, b>0, \alpha>0$$

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

2.12 Yamada Rayleigh et.al model

It attempts to account the testing effort. The mean value function and intensity function are given as

$$m(t) = a\left(1 - exp\left[-r\alpha\left(1 - exp\left[-\frac{\beta t^2}{2}\right]\right)\right]\right), \qquad a > 0, r > 0, \alpha > 0, \beta > 0$$

$$\lambda(t) = ar\alpha\beta t\left(exp\left[-r\alpha\left(1 - exp\left[-\frac{\beta t^2}{2}\right]\right)\right]\right) * exp[-\frac{\beta t^2}{2}], \qquad a > 0, r > 0, \alpha > 0, \beta > 0$$

where a is the expected total number of faults to be eventually detected and α is constant fault introduction rate. r and β are constants.

2.13 Yamada Imperfect Debugging Model 2

It assumes constant introduction rate α and constant fault detection rate b. The mean value function and intensity function of model is given as

$$m(t) = a * (1 - exp[-bt]) * \left(1 - \frac{\alpha}{b}\right) + \alpha at, \qquad a > 0, b > 0, \alpha > 0$$

$$\lambda(t) = a * b(exp[-bt]) * \left(1 - \frac{\alpha}{b}\right) + \alpha a, \qquad a > 0, b > 0, \alpha > 0$$

2.14 Yamada Exponential Model

It attempts to account testing effort and the mean value function and intensity function given below as

$$m(t) = a * (1 - exp[-r\alpha(1 - exp[-\beta t])]), \qquad a > 0, b > 0, \alpha > 0, \beta > 0$$

$$\lambda(t) = ar\alpha(exp[-r\alpha(1 - exp[-\beta t])]) * exp[-\beta t], \qquad a > 0, b > 0, \alpha > 0, \beta > 0$$

where a is the expected total number of faults to be eventually detected and α is constant fault introduction rate. r and β are constants.

2.15 Pham–Nordmann–Zhang (P–N–Z) model

It assumes that introduction rate is a linear function of testing time, and the fault detection rate function is non-decreasing with an inflexion S-shaped model. The mean value function of this model given as

$$m(t) = \frac{a * (1 - exp[-bt]) * \left(1 - \frac{\alpha}{b}\right) + \alpha at}{1 + \beta * exp[-bt]}, \qquad a > 0, b > 0, \alpha > 0, \beta > 0$$

$$\lambda(t) = \frac{ab * exp[-bt](1 - \alpha/b) + \alpha}{1 + \beta \mathrm{exp}[-bt]} + \frac{ab\beta * exp[-bt]\left(1 - \frac{\alpha}{b}\right)(1 - exp[-bt]) + \alpha t}{(1 + \beta * exp[-bt])^2},$$
$$a > 0, b > 0, \alpha > 0, \beta > 0$$

where a is the expected total number of faults to be eventually detected and α is constant fault introduction rate. β is constant.

2.16 Pham–Zhang (P–Z) model

It assumes that introduction rate is exponential function of the testing time, and the fault detection rate is non decreasing with an inflexion S-shaped model. The mean value function and intensity function of this is given as

$$m(t) = \frac{1}{(1+\beta * exp[-bt])} * \left((c + a) * (1 - exp[-bt]) - \frac{ab}{b-\alpha} * (exp[-\alpha t] - exp[-bt])\right), \quad a>0,\ b>0,\ c>0,\ \alpha>0,\ \beta>0$$

$$\lambda(t) = \frac{b(c+a)(1+\beta)\, exp[-bt] - [b * exp[-bt](1+\beta * exp[-\alpha t]) - \alpha * exp[-\alpha t](1+\beta * exp[-bt])]}{(1+\beta * exp[-bt])}, \quad a>0,\ b>0,\ c>0,\ \alpha>0,\ \beta>0$$

where a is the expected total number of faults to be eventually detected and α is constant fault introduction rate. β and c are constants.

2.17 Pham Zhang IFD model

It assumes constant initial fault content function and imperfect fault detection rate combining the fault introduction phenomenon. The mean value of function and intensity function of this model given as

$$m(t) = a - a * exp[-bt] * (1 + (b + d) * t + bdt^2), \qquad a > 0, b > 0, d > 0$$

$$\lambda(t) = a * exp[-bt] * (bt(b - d) + d((tb)^2 - 1), \qquad a > 0, b > 0, d > 0$$

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

2.18 Zhang-Teng-Pham model

It assumes constant fault introduction rate, and the fault detection rate function is nondecreasing with an infle-xion S shape model. The mean value function and intensity function of this model is given as

$$m(t) = \frac{a}{p - \beta} * \left(1 - \frac{(1 + \alpha) * exp[-bt]}{1 + \alpha * exp[-bt]}\right)^{\frac{c}{b}(p-\beta)}, \qquad a > 0, b > 0, c > 0, p > 0, \alpha > 0, \beta \geq 0$$

$$\lambda(t) = \frac{ac}{(1 + \alpha * \exp[-bt])}\left[\frac{(1 + \alpha) * exp[-bt]}{1 + \alpha * \exp[-bt]}\right]^{\frac{c}{b}(p-\beta)} , a > 0, b > 0, c > 0, p > 0, \alpha > 0, \beta \geq 0$$

Where a is the expected total number of faults to be eventually detected and α is constant fault introduction rate. c, p and β are constants.

III. Comparison Criteria

A model can be judged according to its ability to reproduce the observed behavior of the software, and to predict the future behavior of the software from the observed failure data. To investigate the effectiveness of software reliability growth models, set a of comparison criteria is proposed to compare models quantitatively. The comparison criteria which we used are described as follows.

1. The Bias is defined as:

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

It is the sum of the difference between the estimated curve, and the actual data.

2. The mean square error (MSE) measures the deviation between the predicted values with the actual observations, and is defined as:

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

3. The mean absolute error (MAE) is similar to MSE, but the way of measuring the deviation is by the use of absolute values. It is defined as:

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

4. The mean error of prediction (MEOP) sums the absolute value of the deviation between the actual data and the estimated curve, and is defined as:

$$MEOP = \frac{\sum_{i=1}^{k}|m(t_i) - m_i|}{k - p + 1}$$

5. The accuracy of estimation (AE) can reflect the difference between the estimated numbers of all errors with the actual number of all detected errors. It is defined as:

$$AE = \left|\frac{M_a - a}{M_a}\right|$$

Where M_a and are the actual, and estimated cumulative number of detected errors after the test, respectively.

6. The noise is defined as:

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

7. The predictive-ratio risk (PRR) is defined as:

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

PRR measures the distance of model estimates from the actual data against the model estimate.

8. The variance is defined as:

$$Variance = \sqrt{\frac{\sum_{i=1}^{k}(m_i - m(t_i) - Bias)^2}{k-1}}$$

It is the standard deviation of the prediction bias.

9. The Root Mean Square Prediction Error (RMSPE) is a measure of the closeness with which the model predicts the observation. It is defined as:

$$RMSPE = \sqrt{Variance^2 + Bias^2}$$

10. Rsq can measure how successful the fit is in explaining the variation of the data. It is defined as:

$$R_{sq} = 1 - \frac{\sum_{i=1}^{k}(m_i - m(t_i))^2}{\sum_{i=1}^{k}(m_i - \sum_{j=1}^{k}\frac{m_j}{n})^2}$$

11. The sum of squared errors (SSE) is defined as:

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

12. The Theil statistic (TS) is the average deviation percentage over all periods with regard to the actual values. The closer Theil's Statistic is to zero, the better the prediction capability of the model. It is defined as:

$$TS = 100 * \sqrt{\frac{\sum_{i=1}^{k}(m_i - m(t_i))^2}{\sum_{i=1}^{k}{m_i}^2}}\%$$

In (1)–(12), k represents the sample size of the data set, and p is the number of parameters. The comparison criteria, Root Mean Square Prediction Error (RMSPE) is a combination of the comparison criteria 'bias' and 'variance'. The criteria MSE, MAE and MEOP are used to measure the deviation whereas the criteria AE and SSE measure the errors. In order to avoid the replication of the criteria and in order to investigate the effectiveness of software reliability models, a set of seven distinct comparison criteria namely mean absolute error (MAE), accuracy of estimation (AE), noise, predictive-ratio risk (PRR), Root Mean Square Prediction Error (RMSPE), R square (Rsq) and Theil statistic (TS) are proposed to compare models quantitatively.

IV. Our Ranking Methodology

To depart from complexity of the formulation of objective and constraint functions that occur when the mathematical programming model is used in a multi-attributes decision problem, a modest attempt is made in this dissertation to develop a deterministic quantitative model based on weighted mean for the purpose of ranking of software reliability models. Since in this method we are using the weight of each criteria so we will called this method is weighted criteria method. For applying our methodology we use matrix to write the value of criteria for each model. Steps involved in weighted criteria method given as follows:-

1. Criteria Value Matrix

In this matrix each element a_{ij} shows the value of jth criteria of ith model. Let us consider n number of SRGMs with m criteria. This matrix can be given below as:

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

Where $(Amax)_j$ = Maximum value of jth criteria, $(Amin)_j$ = Minimum value of jth criteria and

a_{ij} = Value of jth criteria of ith model.

2. Criteria Rating

As these criterion ratings are different for different software reliability models, hence, the criteria rating matrix differs from model to model. The criteria ratings are determined as under:

Case - I: When smaller value of the criterion represents fitting well to the actual data i.e. is the best value:

$$Criteria\ rating = \frac{Criteria\ Maximum\ Value\ in\ the\ Database - Criteria\ Value}{Criteria\ Maximum\ Value\ in\ the\ Database - Criteria\ Minimum\ Value\ in\ the\ Database}$$

Case - II: When bigger value of the criterion represents fitting well to the actual data i.e. is the best value:

$$Criteria\ rating = \frac{Criteria\ Value - Criteria\ Minimum\ Value\ in\ the\ Database}{Criteria\ Maximum\ Value\ in\ the\ Database - Criteria\ Minimum\ Value\ in\ the\ Database}$$

3. Criteria Weight

Let us consider X_{ij} represent the rating of jth criteria of ith model. Weight of criteria will be W_{ij} = 1- X_{ij}.

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

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

where i = (1, 2, 3......n) and j = (1, 2, 3 ....m)

4. Weight Matrix

The weight matrix can be represented as:

$$Weight\ Matrix = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1m} \\ W_{21} & W_{22} & \cdots & W_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ W_{n1} & W_{n2} & \cdots & W_{nm} \end{bmatrix}$$

5. Weighted Criteria Value Matrix

Weighted criteria value is calculated by multiplying the weight of each criterion with the criteria value i.e. $A_{ij} = W_{ij} * a_{ij}$

$$Weighted\ Criteria\ Value\ Matrix = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{bmatrix}$$

6. Permanent Value of Model

Permanent value of model is the weighted mean value of all criteria. This value is given below:-

$$Z_i = \frac{\sum_{i=1}^{m} A_{ij}}{\sum_{i=1}^{m} W_{ij}}, \quad \text{where i = (1, 2, 3 ...n)}$$

Ranking of models are done the basis of permanent value of model as calculated above. Smaller permanent value of model represents good rank as compare to bigger permanent value of model. So compare all permanent values and provide ranks for each model.

V. Case Study

Case Study 1:- A dataset (DS1) consider in Kapil Sharma et al.[22] having 100 reported defects has been taken from the open literature for evaluation, weighted criteria , and ranking of these sixteen NHPP software reliability growth models based on eight criteria as described above : MSE, MAE, MEOP, AE, Noise, RMSPE, SSE, and TS. The data set, as in Table 1, was collected from a subset of products for four separate software releases at Tandem Computers Company. To avoid confidentiality issues, the number of faults was normalized from 0 to 100, and the amount of testing effort (TE) consumption was proportionally translated into the range (0, 10,000).

Table 1: Tandem computers software failure (ds1)

Weeks	CPU hours	Defects found	Weeks	CPU hours	Defects found
1	519	16	11	6539	81
2	968	24	12	7083	86
3	1430	27	13	7487	90
4	1893	33	14	7846	93
5	2490	41	15	8205	96
6	3058	49	16	8564	98
7	3625	54	17	8923	99
8	4422	58	18	9282	100
9	5218	69	19	9641	100
10	5823	75	20	10000	100

The values of the parameters for these sixteen NHPP SRGMs have been estimated using the least square estimation (LSE) technique, and confidence bounds of 95%. The parameters have been estimated using time weeks because the weekly consumption of testing effort gradually decreased after the t=11 week. The estimated values of the parameters have been provided in Table 2.The values of the eight comparison criteria considered in this case have been obtained using criteria formulas as describe above in section for each model.

Table 2: Parameter estimation of the SGRMs-DS1

Model Name	Least Square Estimation of Parameter
1. Generalized Goel	$a = 68.554$, $b = 7.934{*}10^{-3}$, $c = 0.450$
2. Goel-Okumoto	$a = 169.635$, $b = .057$
3. Gompertz	$a = 151.328$, $b = .125$, $c = .085$
4. Inflection S-shape	$a = 168.717$, $b = .057$, $\beta = 1.024{*}10^{-4}$
5. Logistic Growth	$a = 107.818$, $b = 0.269$, $k = 6.535$
6. Modified Dunae	$a = 208.9$, $b = 64.05$, $k = .86$
7. Musa-Okumoto	$a = 119.538$, $b = .085$
8. Yamada Imperfect Debugging Model 1	$a = 52.124$, $b = .214$, $c = 0.069$
9. Yamada Rayleigh	$a = 93.3$, $b = 0.167$, $c = .0333$
10. Delayed S-shped	$a = 85.117$, $b = 0.339$
11. Yamada Imperfect Debugging Model 2	$a = 9.532$, $b = 8.241$, $c = .685$
12. Yamada Exponential	$a = 234.769$, $\alpha = 0.183$, $\beta = 0.021$
13. P-N-Z Model	$a = 50.172$, $b = 0.212$, $\alpha = 0.1$, $\beta = 0.001$
14. P-Z Model	$a = 166.331$, $b = 0.057$, $c = 864$, $\alpha = 727782$, β $=1{*}10^{-4}$
15. Pham Zhang IFD	$a = 84.364$, $b = .344$, $d = 5.12{*}10^{-10}$
16. Zhang-Teng-Pham	$a = 18.973$, $b = 7.41{*}10^{-7}$, $c = 0.5$, $\alpha = 4.876$, $\beta = 0.877$

It is observed that the ranking of the SRGMs varies with respect to the selection of criteria set. Actually different criteria-set have different collective impact on the reliability of software. In the present method, each comparison criteria is considered as an individual selection attribute for the evaluation and comparison of non-homogeneous Poisson process SRGMs. To avoid this problem, we apply weighted criteria methodology to rank the SRGMs based on all these eight criteria taken collectively. Table 4 shows the rank calculated by weighted criteria methodology.

Table 3: Weighted value of criteria

Model/ Criteria	1. MSE	2. MAE	3. MEOP	4. AE	5. Noise	6. RMSPE	7. SSE	8. TS	Sum
1. Generalized Goel	85.559174	8.3712745	24.502	0.979	0.3855551	13.288572	1723.8412	11.746518	1868.6733
2. Goel-Okumoto	0.4522375	0.2425768	0.7994319	0.0179614	0.1668115	0.3354626	9.3203621	0.6667061	12.00155
3. Gompert	3.3483416	0.7349187	3.8560346	0.0483155	0.2154421	1.762726	69.462582	2.0961485	81.524509
4. Inflection S-shape	0.3988041	0.2743504	0.7667044	0.0174013	0.1693321	0.320838	8.8303933	0.6458803	11.423704
5. Logistic Growth	0	0	0	0	1.1789505	0	0	0	1.1789505
6. Modified Duane	0.5341524	0.3155596	0.9377699	0.3140236	0	0.3982675	11.673953	0.761004	14.93473
7. Musa okumoto	0.9243209	0.3510711	1.3071262	0.0284421	0.1127949	0.5709328	18.973977	1.0085923	23.277258
8. Yamada Model1	79.25392	3.5435993	17.751461	0.3059421	0.1469802	8.4548331	1597.2604	11.286438	1718.0035
9. Yamada Rayleigh	17.319847	3.5580808	19.073602	0.7448176	5.479	2.4479268	352.0871	5.0675273	405.7779
10. Delayed S-shape	5.7948583	1.6931753	5.8079148	0.0182446	2.7691018	3.0774845	116.96492	2.7917467	138.91745
11. Yamada Model2	21.930107	1.8251638	9.6283105	0.1458991	2.2369108	4.5833593	445.00414	5.7434553	491.09735
12. Yamada Exp.	1919.854	47.654	2.395486	0.7483991	0.019004	94.054	38457.074	58.409	40580.208
13. P-N-Z Model	3.6180498	0.9118589	3.1087423	0.061985	0.2069774	1.4647943	76.734366	2.2147982	88.321572
14. P-Z Model	0.3576827	0.3837216	0.7884493	0.4646674	0.1675298	0.3306356	9.1589486	0.6598031	12.311438
15. Pham Zhang IFD	6.5402065	2.06796	6.3477379	0.0202876	2.8348341	3.3211573	134.29528	3.0106789	158.43814
16. Zhang-Teng-Pham	0.358311	0.3839079	0.7895821	0.4991996	0.0037512	0.331376	9.1731308	0.6604696	12.199728

Table 4: Model permanent value & ranking

Model	Sum of Weight	Sum of Weighted Value	Model Value	Model Rank
1. Generalized Goel	3.885105	1868.673	480.984	14
2. Goel-Okumoto	0.652989	12.00155	18.37941	6
3. Gompert	1.253868	81.52451	65.01843	8
4. Inflection S-shape	0.648525	11.4237	17.6149	5
5. Logistic Growth	0.45292	1.178951	2.603	1
6. Modified Duane	0.97566	14.93473	15.3073	4
7. Musa okumoto	0.778764	23.27726	29.89	7
8. Yamada Model1	2.925656	1718.004	587.2199	15
9. Yamada Rayleigh	3.627201	405.7779	111.8708	12
10. Delayed S-shape	1.947422	138.9174	71.33404	10
11. Yamada Model2	2.521584	491.0974	194.7575	13
12. Yamada Exp.	6.204357	40580.21	6540.598	16
13. P-N-Z Model	1.249445	88.32157	70.68863	9
14. P-Z Model	1.233008	12.31144	9.98488	2
15. Pham Zhang IFD	2.029449	158.4381	78.06954	11
16. Zhang-Teng-Pham	1.113345	12.19973	10.95773	3

Case Study II: - In order to evaluate and compare the models, failure data considered by Kapil Sharma et al. [22] is used in this research paper. As reported by the researchers, this data set is from the testing process on a middle-size software project. Table 5 shows failure data. First column present failure time in weeks and second column presents cumulative number of failures. The values of the parameters for these sixteen NHPP SRMs have been estimated using the MLE technique and confidence bounds of 95%. The estimated values of the parameters have been provided in Table 6.

Table 5: Software failure data set (DS2)

Week	Cumulative Faults	Week	Cumulative Faults	Week	Cumulative Faults
1	15	8	134	15	179
2	35	9	139	16	182
3	60	10	141	17	184
4	74	11	148	18	185
5	94	12	149	19	187
6	102	13	157	20	191
7	114	14	173	21	192

The values of the seven comparison criteria (AE, Noise, Rsq, TS, PRR, RMSPE and MAE) considered in this research paper has been obtained. Using relevant equations, the estimated and optimal values of the parameters are given in Table 6. From the comparison of rankings of the sixteen SRMs based on the values of all these seven criteria, it is shown that the ranking of the SRMs varies with respect to the selection of criteria. No single model is best suitable for all comparison criteria. In order to avoid this problem it is proposed to apply weighted criteria methodology to analyze the performance and rank the SRMs based on all these seven criteria taken collectively. Table 8 clearly shows the rank calculated by weighted criteria methodology.

Table 6: Parameter estimation of SGRMs for DS2

Model Name	Maximum Likelihood Estimation of Parameter
1. Generalized Goel	$a = 185.36$, $b = 8.0*10^{-4}$, $c = 3.1005$
2. Goel-Okumoto	$a = 215.763$, $b = .108$
3. Gompertz	$a = 191.787$, $b = .242$, $c = 5.972*10^{-2}$
4. Inflection S-shape	$a = 203.307$, $b = 0.155$, $\beta = 0.524$
5. Logistic Growth	$a = 188.349$, $b = 0.332$, $k = 7.215$
6. Modified Dunae	$a = 237.581$, $b = 40.437$, $k = 4.096$
7. Musa-Okumoto	$a = 113.003$, $b = .230$
8. Yamada Imperfect Debugging Model 1	$a = 128$, $b = .189$, $c = 2.467*10^{-2}$
9. Yamada Rayleigh	$a = 307.2$, $b = 4.8*10^{-2}$, $c = 3.301*10^{-2}$
10. Delayed S-shped	$a = 190.796$, $b = 0.296$
11. Yamada Imperfect Debugging Model 2	$a = 128$, $b = 0.191$, $c = 3.255*10^{-2}$
12. Yamada Exponential	$a = 307.2$, $\alpha = 6.4*10^{-2}$, $\beta = 0.154$
13. P-N-Z Model	$a = 128$, $b = 0.122$, $\alpha = 0.1$, $\beta = 0.001$
14. P-Z Model	$a = 128$, $b = 0.122$, $c = 81$, $\alpha = 727782$, $\beta = 1*10^{-4}$
15. Pham Zhang IFD	$a = 190.795$, $b = .296$, $d = 1.0*10^{-5}$
16. Zhang-Teng-Pham	$a = 186.35$, $b = 5.223*10^{-2}$, $c = 0.81$, $\alpha = 4.876$, $\beta = 0.877$

Table 7: Weighted value of criteria

Model/ Criteria	1. AE	2. Noise	3. Rsq	4. TS	5. PRR	RMSPE	7. MAE	Sum
1. Generalized Goel	0.001256	15.0034	0.144918	20.63349	11135.07	45.32096	17.18817	11233.36
2. Goel-Okumoto	5.92E-05	0.151702	0.992595	0.004291	6.48E-07	0.003126	0	1.151774
3. Gompert	0.000343	0.666412	0.972526	0.234667	1.59E-06	0.157918	0.183739	2.215607
4. Inflection S-shape	9.48E-07	0.222731	0.993054	0	0	0	0.006074	1.221859
5. Logistic Growth	0.000675	1.180403	0.953858	0.487415	1.04E-05	0.337323	0.462437	3.422122
6. Modified Duane	0.000205	0.019034	0.604411	6.904105	9.33E-05	22.75437	9.863576	40.1458
7. Musa Okumoto	0.001492	0.030077	0.969748	0.270658	2.31E-06	0.398003	0.255214	1.925194
8. Yamada Model1	0.00043	0.012816	0.888677	1.489524	6.31E-06	4.222381	1.963419	8.577253
9. Yamada Rayleigh	0.925064	3.750663	0.779176	3.399596	0.002288	5.406451	4.091774	18.35501
10. Delayed S-shape	0.000427	0.874344	0.953208	0.496619	0.000305	0.338554	0.364488	3.027944
11. Yamada Model2	0	0	0.942603	0.649604	2.94E-06	1.708207	0.781294	4.081711
12. Yamada Exp.	0.884471	0.505561	0.000895	33.01638	0.000555	108.9854	49.10944	192.5027
13. P-N-Z Model	9.11E-05	0.068088	0.981361	0.125541	2.59E-07	0.338237	0.16298	1.676298
14. P-Z Model	6.87E-07	0.23238	0.011518	38.0367	0.009471	119.504	67.87029	225.6644
15. Pham Zhang IFD	0.000427	0.874473	0.953202	0.496697	0.000305	0.338592	0.431239	3.094936
16. Zhang-Teng-Pham	4.45E-05	0.311112	0	34.01094	0.00062	90.48076	48.4929	173.2964

Table 8: Model permanent value & ranking

Model/ Criteria	Sum of weight	Sum of weighted value	Model Value	Model Rank
1. Generalized Goel	4.240354	11233.36	2649.157	16
2. Goel-Okumoto	1.075671	1.151774	1.07075	1
3. Gompert	1.287778	2.215607	1.720488	5
4. Inflection S-shape	1.08897	1.221859	1.122031	2
5. Logistic Growth	1.435293	3.422122	2.384268	8
6. Modified Duane	1.998866	40.1458	20.08429	12
7. Musa okumoto	1.183498	1.925194	1.626698	4
8. Yamada Model1	1.463923	8.577253	5.85909	10
9. Yamada Rayleigh	3.056792	18.35501	6.004665	11
10. Delayed S-shape	1.381429	3.027944	2.191893	6
11. Yamada Model2	1.262489	4.081711	3.233067	9
12. Yamada Exp.	3.883965	192.5027	49.56345	13
13. P-N-Z Model	1.137328	1.676298	1.473892	3
14. P-Z Model	3.196716	225.6644	70.59255	15
15. Pham Zhang IFD	1.387642	3.094936	2.230356	7
16. Zhang-Teng-Pham	2.767109	173.2964	62.62722	14

VI. Conclusion

With the rapid development of computer technology, wide use of computers to control all military and civil systems and increasing demand of high quality software products, software reliability has become the primary concern and it is must to evaluate software reliability accurately and carefully to determine the system reliability. Various models have been proposed to demonstrate software reliability and its dependence on a number of factors related to the product or the software process. Ideally, these models provide a means of characterizing the development process and enable software reliability practitioners to make predictions about the expected future reliability of software under development. Such techniques allow managers to accurately allocate time, money, and human resources to a project with some level of confidence in its reliability. No software reliability growth model is optimal for all contributing criteria. This paper addresses the issue of optimal selection of software reliability growth models. The proposed method is suitable for ranking SRGMs based on a set of criteria taken all together. It is also used to perform sensitivity analysis to identify the most and the least dominating criteria. If we apply weighted matrix method on so many failure data sets then we can achieve the priorities of different criteria. So weighted criteria method uses some weight of each criterion to calculate the overall rank of a model. The method has the flexibility to choose any number of criteria to obtain the final decision. Results can be improved by increasing the number of criteria. In all existing methodologies, calculation procedures are too much complex. The weighted criteria method uses a relatively simple mathematical formulation and straight forward calculation. It is capable of solving complex multi-attributes decision problems, incorporating quantitative factors.

References

[1] Ajeet Kumar Pandey & N. K. Goyal, “A Fuzzy Model for Early Software Fault Prediction Using Process Maturity and Software Metrics”, International Journal of Electronics Engineering, 1(2), 2009, pp. 239-245

[2] J. D. Musa and K. Okumoto, “A Logarithmic Poisson Execution Time Model for Software Reliability Measurement”, Bell Laboratories, Whippany, N. J. 07981

[3] Xiaolin Teng and Hoang Pham, “A New Methodology for Predicting Software Reliability in the Random Field Environments”, IEEE Transactions On Reliability, Vol. 55, No. 3, September 2006

[4] BAI Cheng-Gang, Jiang Chang-Hai, & CAI Kai-Yuan, “A Reliability Improvement Predictive Approach to Software Testing with Bayesian Method”, Proceedings of the 29th Chinese Control Conference, July 29-31, 2010, Beijing, China

[5] Xiaolin Teng, & Hoang Pham, “A Software-Reliability Growth Model for N-Version Programming Systems”, IEEE Transactions On Reliability, Vol. 51, No. 3, September 2002

[6] Kazuhira Okumoto, " A Statistical Method for Software Quality Control", IEEE Transactions On Software Engineering, Vol. Se-1L, No. 12, December 1985

[7] Chin-Yu Huang, Michael R. Lyu & Sy-Yen Kuo, "A Unified Scheme of Some Non-homogenous Poisson Process Models for Software Reliability Estimation", IEEE Transactions On Software Engineering, Vol. 29, No. 3, Page 261-270, March 2003

[8] Roger C. Cheung, "A User-Oriented Software Reliability Model", IEEE Transactions On Software Engineering, Vol. Se-6, No. 2, Page 118-126, March 1980

[9] Hoang Pham, "An Imperfect-debugging Fault-detection Dependent-parameter Software", International Journal of Automation and Computing, 04(4), October 2007, 325-328, DOI: 10.1007/s11633-007-0325-8

[10] Xuemei Zhang, Xiaolin Teng, & Hoang Pham, "Considering Fault Removal Efficiency in Software Reliability Assessment", IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 33, No. 1, January 2003

[11] Anthony Iannin et al., "Criteria for Software Reliability Model Comparisons", IEEE Transactions On Software Engineering, Vol Se-10, No. 6, Page 687-692, November 1984

[12] Chin-Yu Huang and Michael R. Lyu, "Estimation and Analysis of Some Generalized Multiple Change-Point Software Reliability Models", IEEE Transactions On Reliability, Vol. 60, No. 2, Page 498-515, June 2011

[13] Katerina Goˇseva-Popstojanova and Kishor S. Trivedi, "Failure Correlation in Software Reliability Models", IEEE Transactions On Reliability, Vol. 49, No. 1, Page 37-49, March 2000

[14] Shinji Inoue and Shigeru Yamada, "Generalized Discrete Software Reliability Modeling With Effect of Program Size", IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 37, No. 2, Page 170-180, March 2007

[15] Kazuya Shibata, Koichiro Rinsaka and Tadashi Dohi, "Metrics-Based Software Reliability Models Using Non-homogeneous Poisson Processes", 17th International Symposium on Software Reliability Engineering (ISSRE'06)

[16] Hoang Pham and Xuemei Zhang, "NHPP software reliability and cost models with testing coverage", European Journal of Operational Research 145 (2003) 443–454

[17] Michael R. Lyu, Sampath Rangarajan and Aad P. A. van Moorsel, "Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development", IEEE Transactions On Reliability, Vol. 51, No. 2, Page 183-193, June 2002

[18] K.B.P.L.M. Kelani Bandara et al., "Optimal Selection of Failure Data for Reliability Estimation Based on a Standard Deviation Method", Second International Conference on Industrial and Information Systems, ICIIS 2007, 8 – 11 August 2007, Sri Lanka

[19] RajPal Garg, Kapil Sharma, Rajive Kumar, and R. K. Garg, "Performance Analysis of Software Reliability Models using Matrix Method", World Academy of Science, Engineering and Technology July, 2010

[20] Sultan Aljahdali and Alaa F. Sheta, "Predicting the Reliability of Software Systems Using Fuzzy Logic", 2011 Eighth International Conference on Information Technology: New Generations

[21] Ralf H. Reussner, Heinz W. Schmidt and Iman H. Poernomo, "Reliability prediction for component-based software architectures", The Journal of Systems and Software 66 (2003) 241–252

[22] Kapil Sharma,Rakesh Garg, C. K. Nagpal, and R. K. Garg, "Selection of Optimal Software Reliability Growth Models Using a Distance Based Approach", IEEE Transactions On Reliability, Vol. 59, No. 2, Page 266-277, June 2010

[23] Norman F. Schneidewind, "Software Reliability Model with Optimal Selection of Failure Data", IEEE Transactions On Software Engineering, Vol. 19, No. 11, Page 1095-1105, November 1993

[24] Dr. Ajay Gupta, Dr. Digvijay Choudhary and Dr. Suneet Saxena, "Software Reliability Estimation using Yamada Delayed S Shaped Model under Imperfect Debugging and Time Lag", International Journal of Computer Applications (0975-8887) Volume 23– No.7, June 2011

[25] Shaik.Mohammad Rafi et al., "Software Reliability Growth Model with Logistic-Exponential Test-Effort Function and Analysis of Software Release Policy", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 387-399

Mohd Anjum was born in 1986 in Bareilly, India. He received his B.Tech degree in Computer Science and Engineering from Aligarh Muslim University in 2010. He received his Master degree in Software Engineering from the same University in 2012. Now he is working as an Assistant Professor (Ad-hoc basis) in AMU. His area of interests includes Software engineering, Automata Theory and Networking.

Md. Asraful Haque was born in 1985 in West Bengal, India. He received his Master degree in Computer Science and Engineering (Specialization-Software Engineering) from Aligarh Muslim University. Presently he is an Assistant Professor (Ad-hoc basis) in Aligarh Muslim University. He has more than three years of teaching experience. His area of interests includes Software engineering, Operating Systems, Data Structure, Image Processing and Computer Security. He has authored several papers in various International Journals and Conference proceedings.

Nesar Ahmad is presently working as a Professor in the Department of Computer Engineering, Aligarh Muslim University, India. He has nearly twenty years of teaching experience. Nesar Ahmad obtained his B.Sc (Engg) degree in Electronics & Communication Engineering from Bihar College of Engineering, Patna (Now NIT, Patna) in 1984 and M.Sc (Information Engineering) degree from City University, London, U.K., in 1989. He received his Ph.D degree from Indian Institute of Technology (IIT), Delhi, India, in 1993. Earlier he worked as a Senior Scientific Officer in Microprocessor Applications Program at IIT Delhi. He was with King Saud University, Riyadh during 1997-99 as an Assistant Professor. Before joining AMU, he worked as an Assistant Professor in the Department of Electrical Engineering, IIT Delhi till December 2004. His current research interests mainly include Soft Computing, Web Intelligence and E-Learning. He has authored many papers in various International Journals and Conference proceedings.

How to cite this paper: Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad,"Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value", International Journal of Information Technology and Computer Science(IJITCS), vol.5, no.2, pp.1-14, 2013.DOI: 10.5815/ijitcs.2013.02.01