Special Issue: SAICSIT '99
Contents

Preface
P. Machanick ............................................................... 1

Research Articles
Active Learning: Issues and Challenges for Information Systems and Technology
RD Quilling, GJ Erwin and O Petkova ................................. 5

A Generic Modelling Framework for Interactive Authoring Support Environments
Paula Kotze ............................................................... 15

O Petkova and JD Roode ................................................... 26

An Information-Theoretic Semantics for Belief Change
T Meyer ................................................................. 33

A Complexity Metrics Model for Software Correction
A Törn, T Andersson and K Enholm ....................................... 40

A Conceptual Design for High-Volume Data Processing of Warehouse Database into Multidimensional Database
Paisarn Trakulsuk and Vichit Avatchanakorn ......................... 49

A Pragmatic Approach to Bitemporal Databases: Conceptualization, Representation and Visualisation
Chiyaba Njovu and WA Gray .............................................. 58

A Building Recognition System
SP Levitt and B Dwolatzky .................................................. 68

Computer Programming and Learning to Write
John Barrow .............................................................. 77

Co-operating to Learn using JAD Technologies
TA Thomas ................................................................. 87

Critical Success Factors for the Implementation of DSS at a Selection of Organisations in Kwazulu/Natal
URF Averweg and GJ Erwin .............................................. 95

Enhancing the Predictability of Two Popular Software Reliability Growth Models
Peter A Keiller and Thomas A Muzzuchi ............................... 105
Generalised Unification of Finite Temporal Logic Formulas
Scott Hazelhurst .............................................................. 110

Harmonizing Global Internet Tax: A Collaborative Extranet Model
E Lawrence and B Garner ...................................................... 119

Improving Object Oriented Analysis by Explicit Change Analysis
Lui Yu, Siew Chee Kong, Yi Xun and Miao Yuan ....................... 128

Reconciling the Needs of New Information Systems Graduates and Their Employers in Small, Developed Countries
Rodney Turner and Glenn Lowry ........................................... 136

Shortest Delay Scheduling Algorithm for Lossless Quality Transmission of Stored VBR Video under Limited Bandwidth
Fei Li, Yan Liu, Jack Yiu-Bun Lee and Ishfaq Ahmad .................... 146

Software Croma Keying in an Immersive Virtual Environment
Frans van der Berg and Veli Lalioti ........................................ 155

Some Automata-Theoretic Properties of  \( \bigcap \)-NFA
Lynette van Zijl and Andries PJ van der Walt .......................... 163

Statistical Analysis of an Automated Computer Architecture Learning Environment
JT Waldron, J Horgan and G Keogh ........................................ 168

The CILT Multi-Agent Learning System
Herna L Viktor ...................................................................... 176

The Development of a Generic Framework for the Implementation of Cheap, Component-Based Virtual Video-Conferencing System
Soteri Panagou and Shaun Bangay .......................................... 185

The Role of Experience in User Perceptions of Information Technology: An Empirical Examination
Meliha Handzic and Graham Low ............................................. 194

What are Web Sites Used for: Cost Savings, Revenue Generating or Value Creating?
Man-Ying Lee ....................................................................... 201

New Ideas Papers

Approaches to Video Transmission over GSM Networks
Bing Du and Anthony Maeder ................................................. 210

From Information Security Baselines to Information Security Profiles
Rossouw von Solms and Helen van der Haar ................................ 215

Grounded Theory Methodology in IS Research: Glaser versus Strauss
J Smit .................................................................................. 219

Introducing a Continuum of Abstraction-Led Hierarchical Search Techniques
Robert Zimmer and Robert Holte ............................................. 223
Multimedia as a Positive Force to Leverage Web Marketing, with Particular Reference to the Commercial Sector
Stan Shear ................................................................. 229

Understanding HCI Methodologies
Peter Warren .......................................................... 234

Electronically Published Papers

<table>
<thead>
<tr>
<th>Experience Papers</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Java Client/Server System for Accessing Arbitrary CANopen Fieldbus Devices via the Internet</td>
<td>239</td>
</tr>
<tr>
<td>Dieter Bühler, Gerd Nusser, Gerhard Gruhler and Wolfgang Küchlin</td>
<td></td>
</tr>
<tr>
<td>An Object-Oriented Framework for Rapid Client-side Integration of Information Management Systems</td>
<td>244</td>
</tr>
<tr>
<td>Ralf-Dieter Schimkat, Wolfgang Küchlin and Rainer Krautter</td>
<td></td>
</tr>
<tr>
<td>Distributed Operating Systems: A Study in Applicability</td>
<td>249</td>
</tr>
<tr>
<td>Jürgen Prange and Judith Bishop</td>
<td></td>
</tr>
<tr>
<td>Formal Verification with Natural Language Specifications: Guidelines, Experiments and Lessons so far</td>
<td>253</td>
</tr>
<tr>
<td>Alexander Holt</td>
<td></td>
</tr>
<tr>
<td>Introducing Research Methods to Computer Science Honours Students</td>
<td>258</td>
</tr>
<tr>
<td>Vashti Galpin, Scott Hazelhurst, Conrad Mueller and Ian Sanders</td>
<td></td>
</tr>
<tr>
<td>Visualising Eventuality Structure</td>
<td>264</td>
</tr>
<tr>
<td>ST Rock</td>
<td></td>
</tr>
</tbody>
</table>

Electronically Published Papers
SAICSIT'99
South African Institute of Computer Scientists and Information Technologists
Annual Research Conference 17-19 November 1999
Prepare for the New Millennium
Is there life after y2k?
Mount Amanzi Lodge, Hartebeespoort, South Africa

Sponsors

http://www.cs.wits.ac.za/
PHDS
research/PHDS.html
Think different.
Preface

Philip Machanick, Overall Chair: SAICSIT'99

Running SAICSIT'99, the annual research conference of the South African Institute for Computer Scientists and Information Technologists, has been quite an experience.

SAICSIT represents Computer Science and Information Systems academics and professionals, mainly those with an interest in research. When I took over as SAICSIT president at the end of 1998, the conference had not previously been run as an international event. I decided that South African academics had enough international contacts to put together an international programme committee, and a South African conference would be of interest to the rest of the world.

I felt that we could make this transition at relatively low cost, given that we could advertise via mailing lists, and encourage electronic submission of papers (to reduce costs of redistributing papers for review).

The first prediction turned out to be correct, and we were able to put together a strong programme committee.

As a result, we had an unprecedented flood of papers: 100 submitted from 21 countries. As papers started to come in, it became apparent that we needed more reviewers. It was then that the value of the combination of old-fashioned networking (people who know people) and new-fashioned networking (the Internet) became apparent. While the Internet made it possible to convert SAICSIT into an international event at relatively low cost, the unexpected number of papers made it essential to find many additional reviewers on short notice. Without the speed of e-mail to track people down and to distribute papers for review, the review process would have taken weeks longer, and it would have been much more difficult to track down as many new reviewers in so little time.

Even so, the number of referees who were willing to help on short notice was a pleasant surprise.

The accepted papers cover an interesting range of subjects, from management-interest Information Systems, to theoretical Computer Science, with subjects including database, Java, temporal logic and implications of e-commerce for tax.

In addition, we were very fortunate in being able to invite the president of the ACM, Barbara Simons as a keynote speaker. Consequently, the programme for SAICSIT'99 should be very interesting to a wide range of participants.

We were only able to find place in the proceedings for 36 papers out of the 100 submitted, of which only 24 are full research papers. While this number of papers is in line with our expectation of how many papers would be accepted in each category, we did not have a hard cut-off on the number of papers, but accepted all papers which were good enough, based on the reviews. Final selection was made by myself as Programme Chair, and Derrick Kourie, as editor of the South African Computer Journal. Additional papers are published via the conference web site.

We believe that we have put together a quality programme, and hope you will agree.

Acknowledgments

I would like to thank the South African Computer Journal production team, Andries Engelbrecht and Herna Viktor, respectively from the Department of Computer Science and Informatics, University of Pretoria, for their work on producing the proceedings.

The reviewers listed overleaf did an excellent job: many wrote very detailed reports, sometimes after being called in on very short notice. Inevitably, there were some glitches resulting from the unexpected workload, but the buck stops with the programme chair: I promise to do better next time.

I would also like to thank my own department for putting up with the extra work and expense that running a conference entails. I tried not to burden them with too much extra work, but our secretaries, Zahn Gowar and Leanne Reddy, inevitably had to take on some extra work. John Ostrowick provided valuable assistance with design of our web pages and call for papers poster. Carol Kernick, who handles our finances and membership records, did a fine job of keeping up with the demands of the conference.

Finally, I would like to thank our sponsors, whose contribution made this conference been possible:

- PricewaterhouseCoopers – sponsored generous prizes and the conference banquet
- National Research Foundation (NRF) – provided financial support
- University of the Witwatersrand – provided financial support
- Programme for Highly Dependable Systems, University of the Witwatersrand – provided financial support
- Standard Bank – provided financial support
Editorial

• Apple Computer – provided equipment for the conference
• Qualica – provided technical support including helping with the conference web site

Web Site

For more information about SAICSIT, including a pointer to the conference site, see <http://www.saicsit.org.za>.

Referees

• Department of Computer Science, University of Pretoria
  – Derrick Kourie
  – Bruce Watson
  – Vali Lalioti
  – Andries Engelbrecht
  – Ivan Mphahlele
• German National Research Center for Information Technology - GMD
  – Gernot Goebbels
• School of Math Stat and IT, Natal University Pietermaritzburg
  – Peter Warren
• Graduate School of Business, University of Cape Town
  – Kurt April
• CSIR
  – James Jardine
• Department of Computer Science, Rhodes University
  – Shaun Bangay
  – Peter Clayton
  – John Ebden
  – Richard Foss
  – George Wells
  – Peter Wentworth
• Information Technology Division, Rhodes University
  – Caro Watkins
• Department of Information Systems, Rhodes University
  – Brenda Mallinson
• Department of Informatics, University of Pretoria
  – Carina de Villiers
  – Herna Viktor
  – Niek du Plooy
  – Elsje van Rooyen
  – Machdel Matthee
  – Alan Abrahams
  – Jackie Phahlamohlaka
• School of Information Technology & Engineering, University of Ottawa
  – Stan Szpakowicz
  – Dwight Makaroff
• Department of Information Systems, Victoria University
  – Glenn Lowry
  – Peter Shackleton
  – Tas Adam
  – Alastair Wallace
  – Stephen Burgess
  – Dave Burgess
  – Julie Fisher
  – Jerzy Lepa
  – Geoff Sandy
  – Rod Turner
  – Alastair Wallace
  – Andrew Wenn
• Peninsula School of Computing and Information Technology, Monash University
  – Ainslie Ellis
• School of Information Management and Systems, Monash University
  – Angela Cardone
• School of Computer Science & Software Engineering, Monash University
  – Dianne Hagan
  – Judy Sheard
• School of Management Information Systems, Deakin University
  – John Lamp
  – David Mackay
  – Philip Joyce
Editorial

- Chris Brew
- Alexander Holt

• Department of Computer Science and Engineering, Helsinki University of Technology
  - Tommi Junttila
  - Nisse Husberg

• Technical University of Budapest, Department of Measurement and Information Systems
  - Andras Pataricza

• NASA-Goddard Space Flight Center
  - Nigel Ziyad

• Department of Computer Science & Information Systems, University of Port Elizabeth
  - Janet Wesson
  - Leon Nicholls

• Computer Technology Department, Indiana University Purdue University Indianapolis
  - Tim Price

• Department of Electrical Engineering and Computer Science, The University of Michigan
  - Trevor Mudge

• Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
  - David Forsyth

• School of Computer Science, University of Birmingham
  - Mark Ryan

• Faculty of Mathematics, Computer Science, Physics & Astronomy, University of Amsterdam
  - Carlos Areces

• Department of Computer Science, Universidade do Vale do Rio dos Sinos - UNISINOS Rio Grande do Sul
  - Marcelo Walter

• School of Information Technology & Mathematical Sciences, University of Ballarat
  - Binh Pham
Enhancing the Predictability of Two Popular Software Reliability Growth Models

Peter A Keiller and Thomas A Mazzuchi

Abstract

In this paper, enhancement of the performance of both the Musa-Okumoto and the Goel-Okumoto Software Reliability Growth models are investigated using various smoothing techniques. The method of parameter estimation for the models is the Maximum Likelihood Method. The evaluation of the performance of the models is judged by the relative error of the predicted number of failures over future time intervals relative to the number of failures eventually observed during the interval. The use of data analysis procedures utilizing the Laplace Trend test are investigated. These methods test for reliability growth throughout the data and establish "windows" that censor early failure data and provide better model fits. The research showed conclusively that the data analysis procedures resulted in improvement in both models predictive performance for 41 different sets of software failure data collected from software development labs in the United States and Europe.

Keywords: Software Reliability, Maximum Likelihood Method, Laplace Trend Test

Computing Review Category: D.2.4

1 Introduction

It is a known fact that when software is put in service to perform some function it may not perform as desired. This is termed a failure. These failures may be due to errors, ambiguities, oversights or misinterpretations in the specifications the software is developed to satisfy. Carelessness or incompetence in writing code, inadequate testing, incorrect or unexpected usage of the software, may also contribute to the problems. In software failure behavior, statistical models and analysis can be used to investigate various aspects of software and its failure at the different levels of detail. The statistical models used in making predictions about the quality of software are termed software reliability growth models. Research has shown [9, 10, 18] that the majority of these software reliability growth models either consistently overestimate or underestimate the quality of the software. This paper investigates the potential improvement of model fitting to software failure data using "windows" developed utilizing reliability growth techniques.

2 Software reliability modeling

There have been over forty documented software reliability models reported in the published literature [4]. Readers interested in the history and development of software reliability modeling with a more detailed account of the models associated assumptions are directed to [2, 12, 14, 18, 20].

Raw data available to the user can be represented as a sequence of failure times (i.e. $t_1, \ldots, t_c$), where $t_c$ is the 'current time' or as a sequence of inter-failure times (i.e. $x_1, \ldots, x_c$) where $x_i = t_i - t_{i-1}$. Since the failure times are indexed chronologically they can be represented as

$$0 \leq t_1 \leq \ldots \leq t_c.$$  

Because there is uncertainty about when the failures will occur, Equation (1) can be modeled as the realization of a stochastic process $\{T_i; i \geq 0\}$ where $T_i$ are random variables, $t_i$ are real scalars and the process is observed for $0 \leq i \leq t_c$. Another way of modeling the uncertainty is via a stochastic process for the number of cumulative failures in time $t$, $\{N(t); t \geq 0\}$ where

$$N(t) = \max\{i: T_i \leq t\}.$$  

Software reliability models can be classified into two groups: the inter-failure time or times between failure models and the failure count models. Both of these classes of models are based on the failure history of software and are classified according to the nature of the failure process studied. These two classes of models can be further examined based on the inference viewpoint, classical (Frequentist) or Bayesian.

If through the successful redesigns of the system, a noticeable average increase in inter-failure time is observed, then the reliability of the system is said to be improving. We call this phenomenon reliability growth. The stochastic process $\{T_i; i \geq 0\}$ or $\{N(t); t \geq 0\}$ then is called a reliability growth process and can be thought of as consisting of noisy behavior around a smooth trend curve. One
way of modeling this trend curve is with the expected value function for the stochastic process such as

\[ M(t) = E[N(t)], \quad t \geq 0 \]  

(3)

An attractive family of stochastic processes characterized by their mean functions are the non homogeneous Poisson processes (NHPP); Musa and Okumoto [16,18] have promoted the use of these models in software reliability growth modeling. Miller [14] also gives strong theoretical justification for using NHPP's.

Since this process is completely characterized by its mean value function, \( M(t) \), what have developed are families of models with different mean value functions. In this study, the Musa-Okumoto (M/O) model [16] characterized by the mean value function

\[ M(t) = \gamma \ln(1 + \beta t), \quad \gamma, \beta > 0 \]  

(4)

and the Goel-Okumoto (G/O) model [7] characterized by the mean value function

\[ M(t) = \gamma(1 - e^{-\eta t}), \quad \gamma, \eta > 0 \]  

(5)

are investigated.

3 Failure data collection

There are only a few sets of failure data in the open literature. This is mainly due to the confidentiality of this type of data and also due to the unstructured collection procedures used by many organizations. Data (41 different data sets) used in this research is taken from the collections of Musa [17], Brocklehurst [3], and data found in journals and technical reports [6,13,15,19]. These data sets report the inter-failure times for different software development projects.

4 Trend determination

For the different data sets on test, observing the plots of the failure data (inter failure times against the % completion of the test time) only a few groups of data show any obvious improvement as the time increased (meaning longer inter failure times). Reliability growth however can be analyzed through the use of trend tests [5]. Kanoun and Laprie [8] outline one trend test that is widely used.

This test is considered a formal trend analysis test which computes the Laplace factor, \( \mu(t_n) \), for the interval \((0, t_n)\). The statistic is given by:

\[ \mu(t_n) = \frac{\sum_{i=1}^{n} \left( s_i - \frac{t_i}{2} \right)}{t_n \sqrt{\frac{1}{52}}} \]  

(6)

(where \( n \) is the number of failures in \((0, t_0)\), and \( s_i \) the time of occurrence of failure \( i, i = 1, \ldots, n \)). Negative value of the Laplace factor \( \mu(t_0) \) indicates a rate of occurrence of failures decreasing with time. A positive value indicates an increasing failure intensity.

The Laplace factor test in this study is used to construct procedures that are used to derive “windows” that select the “best reliability growth intervals” and trends within the different failure data sets.

The four procedures used during the research are:

Procedure P1: All Data Approach

This is the standard approach used by many researchers. Using all the data within the interval \((0, s_k)\), the model is fitted where \( s_k \) is the current time of occurrence of failure \( k, k = 1, 2, 3, \ldots, n \) and \( n \) is the number of failures in the data set.

Procedure P2: First Growth

This approach constructs the maximum size window that shows reliability growth within the interval. Using all the data within the interval \((0, s_k)\), an interval \((s_k, s_{k-1})\) is selected where \( s_k < s_{k-1} \) and the first indication of reliability growth as outlined by the Laplace test statistic is present. At failure \( k \), the model is fitted using only the data within the interval \((s_k, s_{k-1})\), where \( s_k \) is defined as in Procedure P1. If there is no indication of reliability growth then \( s_k \equiv 0 \).

Procedure P3: Maximum Growth

This approach constructs the window size that has the maximum reliability growth within the interval. Using all the data within the interval \((0, s_k)\), an interval \((s_k, s_{k-1})\) is selected where \( s_k < s_{k-1} \) and the maximum reliability growth as outlined by the Laplace test statistic is indicated. At failure \( k \), the model is fitted using only the data within the interval \((s_k, s_{k-1})\), where \( s_k \) is defined as in Procedure P1. If there is no indication of reliability growth then \( s_k \equiv 0 \).

Procedure P4: Average Growth

This approach is a smoothing technique using the results derive in Procedures P2 and P3. Using all the data within the interval \((0, s_k)\), an interval \((s_k, s_{k-1})\) is selected where \( s_k < s_{k-1} \) and there is an improvement in reliability growth over the previous interval \((0, s_{k-1})\) based on computed dynamic constraints. If there is no improvement over the previous interval’s reliability growth, then the interval will be selected to reflect the best reliability growth within the current interval as outlined by the Laplace factor test. At failure \( k \), the model is fitted using only the data within the interval \((s_k, s_{k-1})\), where \( s_k \) is defined as in Procedure P1. If there is no indication of reliability growth then \( s_k \equiv 0 \).

Following these approaches it can be seen that there is the potential for many more variations of reliability growth “windows” or procedures using the Laplace factor test [11]. In searching the data for unsuspected patterns and abnormalities, the Laplace test shows simple trend changes in the data when applying the different procedures. These changes are noticeable in most of the plots, especially during the first 30% of the total test time of the software. Details of these plots are given in [11].
5 Predictability process

The performance of both models are investigated for each data set to see how well it can predict the number of new failures manifesting during finite future time intervals using each of the four procedures $P_1$ to $P_4$ for model fitting. The experiment is designed as a four step process. For each model, for each procedure $P_i$ and data set $D_k$ where $i=1,...,4$ and $k=1,...,41$:

1. The MLE's of the model parameters are determined based on the first $d$ data points of data set $D_k$.
2. The fitted model is then used to predict the number of failures in $[s_{d,k}, s_{n,k}]$ where
   \[ s_{i,k} = \sum_{m=1}^{d} x_{m,k} \]
   and $x_{m,k}$ is the $m$th data point of data set $D_k$.
3. The model is evaluated using
   (a) predicted errors
   \[ e_{i,k} = \hat{n}_{i,k} - n_{i,k} \]
   (b) relative errors
   \[ re_{i,k} = \frac{\hat{n}_{i,k} - n_{i,k}}{n_{i,k}} \]
   (c) absolute relative errors
   \[ |re_{i,k}| \]
   (d) relative error squared
   \[ (re_{i,k})^2 \]
   where $\hat{n}_{i,k}(n_{i,k})$ is the predicted (actual) number of failures in $[s_{d,k}, s_{n,k}]$.
4. For each data set, the analysis (1) to (3) is performed for $d=10,...,n_{k-1}$. The value ten was selected arbitrarily but is large enough to enable meaningful estimation and small enough to allow for meaningful evaluation of the procedure. The sum of the absolute relative errors (ARE), the sum of the relative errors (RE), and the sum of the relative errors squared (RESQ) are normalized by averaging over the number of predictions $(n_k-10)$ for each model-procedure combination.

6 Comparison procedure

The comparison process of the four procedures is conducted as follows:

1. Hypothesis testing of $RE_{i,j}$, the normalized relative errors of the data set-procedure pair is first conducted to determine if there is any prediction bias when using the procedure.
   \[ H_0 : RE = 0 \quad H_a : RE \neq 0 \]

2. For all unbiased model-procedure pairs, the Sign Test [18] is conducted using the normalized ARE performance measures to determine the spread of the magnitude of the errors.

3. To account for both the bias and variability of the model-procedure pair, the Sign Test is conducted using the normalized RESQ performance measures to confirm the previous steps and also to define the overall winners.

7 Findings

Tables 1 and 2 show the summary of the models' performance measures. Tables 3 to 5 show the results of the hypothesis test (for bias) and the Sign Test for the ARE and RESQ measures for the different procedures. The confidence levels are displayed.

Following the procedure comparison process for the M/O model, the results presented in Table 3 show that only procedures $P_3$ and $P_4$ obtain unbiased predictions. At this point procedures $P_1$ and $P_2$ can be eliminated from consideration. Table 4 outlines the result for the magnitude of the error spread (ARE) based on the Sign Test. This result places $P_3$ and $P_4$ equally. Table 5 outlines the Sign Test result for the combination of bias and variance (RESQ). In this Table, procedures $P_2$, $P_3$, and $P_4$ are all given the edge over $P_1$. The final overall procedure ranking for the Musa-Okumoto model is $\{P_3, P_4, P_2, P_1\}$.

Following the procedure comparison process for the G/O model, the results presented in Table 3 shows that only procedures $P_3$ and $P_4$ obtain unbiased predictions. At this point procedures $P_1$ and $P_2$ can be eliminated from consideration. Table 4 outlines the result for the magnitude of the error spread (ARE) based on the Sign Test. This result places $P_3$ ahead of $P_4$. Table 5 outlines the Sign Test result for the combination of bias and variance (RESQ). In this Table, procedures $P_2$, $P_3$, and $P_4$ are all given the edge over $P_1$. The final overall procedure ranking for the Goel-Okumoto model is $\{P_3, P_4, P_2, P_1\}$.

Tables 6 and 7 show the average percentage improvement in using procedures $P_2$, $P_3$, and $P_4$ over $P_1$ for the three performance measures for the two models.

8 Conclusions

The research was successful in using data analysis trend techniques to improve the predictability performance of the M/O and G/O models. Procedures $P_2$, $P_3$ and $P_4$ perform better than procedure $P_1$ using the bias, absolute relative errors and relative errors squared performance measures.
<table>
<thead>
<tr>
<th>Procedures</th>
<th>Summary Statistics</th>
<th>( P_1 )</th>
<th>( P_2 )</th>
<th>( P_3 )</th>
<th>( P_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average RE</td>
<td>0.751</td>
<td>0.389</td>
<td>0.054</td>
<td>0.060</td>
<td></td>
</tr>
<tr>
<td>Average ARE</td>
<td>0.874</td>
<td>0.546</td>
<td>0.405</td>
<td>0.404</td>
<td></td>
</tr>
<tr>
<td>Average RESQ</td>
<td>2.272</td>
<td>1.413</td>
<td>0.897</td>
<td>0.884</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of M/O Model Performance Measures

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Summary Statistics</th>
<th>( P_1 )</th>
<th>( P_2 )</th>
<th>( P_3 )</th>
<th>( P_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average RE</td>
<td>0.641</td>
<td>0.193</td>
<td>-0.095</td>
<td>-0.084</td>
<td></td>
</tr>
<tr>
<td>Average ARE</td>
<td>0.888</td>
<td>0.521</td>
<td>0.455</td>
<td>0.451</td>
<td></td>
</tr>
<tr>
<td>Average RESQ</td>
<td>2.255</td>
<td>1.196</td>
<td>0.859</td>
<td>0.860</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Summary of G/O Model Performance Measures

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Model</th>
<th>( P_1 )</th>
<th>( P_2 )</th>
<th>( P_3 )</th>
<th>( P_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Musa-Okumoto</td>
<td>****</td>
<td>****</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td></td>
<td>Goel-Okumoto</td>
<td>****</td>
<td>99%</td>
<td>****</td>
<td>99%</td>
</tr>
</tbody>
</table>

**** indicates failure to reject null hypothesis

Table 3: Hypothesis Test for Bias

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Model</th>
<th>( P_1 )</th>
<th>( P_2 )</th>
<th>( P_3 )</th>
<th>( P_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Musa-Okumoto</td>
<td>-</td>
<td>99%</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td></td>
<td>Goel-Okumoto</td>
<td>-</td>
<td>99%</td>
<td>****</td>
<td>95%</td>
</tr>
</tbody>
</table>

- not applicable

**** indicates failure to reject the null hypothesis

Table 4: Sign Test ARE Compared to \( P_1 \)

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Performance measures</th>
<th>( P_2 )</th>
<th>( P_3 )</th>
<th>( P_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RE</td>
<td>48%</td>
<td>93%</td>
<td>92%</td>
</tr>
<tr>
<td></td>
<td>ARE</td>
<td>38%</td>
<td>54%</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>RESQ</td>
<td>38%</td>
<td>61%</td>
<td>61%</td>
</tr>
</tbody>
</table>

Table 6: Average Percentage Improvement Over \( P_1 \) for the M/O Model

<table>
<thead>
<tr>
<th>Procedures</th>
<th>Performance measures</th>
<th>( P_2 )</th>
<th>( P_3 )</th>
<th>( P_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RE</td>
<td>70%</td>
<td>85%</td>
<td>87%</td>
</tr>
<tr>
<td></td>
<td>ARE</td>
<td>41%</td>
<td>48%</td>
<td>48%</td>
</tr>
<tr>
<td></td>
<td>RESQ</td>
<td>47%</td>
<td>62%</td>
<td>62%</td>
</tr>
</tbody>
</table>

Table 7: Average Percentage Improvement Over \( P_1 \) for the G/O Model
Early analysis of the data, shows a fuzzy unpredictable area during the first 30% of the total test time of the failure data sets. This is the time period that accounts for approximately 50% of the discovered errors during the test period. This is also the time period which produces the greatest challenge for using fitted models for future predictions. Keiller and Miller [9] state that there is an effect caused by how the predictions are measured. A model may overestimate the number of future errors by any amount but it may under estimate the number of failures by at most 100%. This means that a few wild overestimates will hurt the average performance more than a wild under estimate (which is probably a more serious error). This observation in a model comparison setting will make the Goel-Oukumoto model (which tends to under estimate) appear better than it really is and the Musa-Oukumoto Model which tends to over estimate appear worse than it really is. They state that a more reasonable error summary might weight an "under estimate by one half" equivalent to an "over estimate by two fold". The evaluation and summarization of performance measures is an open question and is one suggested for future research.

Bendell and Mellor [2] have suggested the use of simple exploratory data analysis techniques and more sophisticated analogues, such as multivariate methods, time series, and proportional hazards modeling, to explore the structure existing in available software reliability data as a basis for future model building. They believe that the software community has reached a "watershed" in software reliability modeling and has to make the decision between improvements to the established approaches to software reliability modeling or a fundamentally different approach which will provide a breakthrough in achieving greater predictive ability. This study has illustrated the promise of the Laplace trend procedures in enhancing the predictive ability for the M/O and G/O software reliability models. Analysis for more general classes of software reliability models is the next logical step.

References


Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal’s readers will be considered for publication under sections marked as Communications of Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript
Manuscripts for review should be prepared according to the following guidelines:

- Use wide margins and 1½ or double spacing.
- The first page should include:
  - the title (as brief as possible)
  - the author’s initials and surname
  - the author’s affiliation and address
  - an abstract of less than 200 words
  - an appropriate keyword list
  - a list of relevant Computing Review Categories
  - Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author’s surname, and should be cited in the text according to the Harvard. References should also be according to the Harvard method.

Manuscripts accepted for publication should comply with guidelines as set out on the SACJ web page, [http://www.cs.up.ac.za/sacj](http://www.cs.up.ac.za/sacj) which gives a number of examples.

SACJ is produced using the \LaTeX\ document preparation system, in particular \LaTeX\ 2e. Previous versions were produced using a style file for a much older version of \LaTeX, which is no longer supported.

Please see the web site for further information on how to produce manuscripts which have been accepted for publication.

Authors of accepted publications will be required to sign a copyright transfer form.

Charges
Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30.00 per final page for contributions which require no further attention. The maximum is R120.00, prices inclusive of VAT.

These charges may be waived upon request of the author and the discretion of the editor.

Proofs
Proofs of accepted papers may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Letters and Communications
Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words. Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book Reviews
Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement
Placement of advertisements at R1000.00 per full page per issue and R500.00 per half page per issue will be considered. These charges exclude specialised production costs, which will be borne by the advertiser. Enquiries should be directed to the editor.