A Bivariate Autoregressive Software Reliability Model

K. Vedavathi
GITAM University
Visakhapatnam – 530 045
AP, India

K. Srinivas Rao
Andhra University
Visakhapatnam – 530 003
AP, India

A. Vinay Babu
JNT University
Hyderabad – 500 085
AP, India

ABSTRACT
Software reliability models play a dominant role in the analysis of failure data for real time command and control software systems. Goel and Okumoto model is a non homogenous Poisson Process software reliability growth model which has gained a lot of importance in software reliability analysis and prediction. The process of parameter estimation is the major drawback of this model because the independent nature of attribute values is considered in estimation. But in real world applications, there are correlations existing among the attributes. Keeping this criterion, a bivariate autoregressive model of order 1 which forms a linear combination of variants namely software faults and test workers is proposed. A numerical illustration is presented to evaluate the performance of the developed model with that of the existing univariate autoregressive models and found that the proposed model outperforms than exiting model in evaluating and predicting software reliability.

Keywords
Software Reliability, Autoregressive Model, Reliability Evolution

1. INTRODUCTION
Traditional software reliability model always assume that the failure process must follow some classical probability distribution, such as Binominal distribution or Poisson distribution. The result of traditional models' assumption is that they only consider the randomness of the input, but ignore other random factors. In practice, the software is dynamic, and it will definitely interact with other factors, such as human, computer etc. In a word, traditional models omitted many random factors viz., testing tools, thought way of testing personnel, his style and experience, imperfect debugging, computer environmental factor, supporting system (such as operating system, compiling program, database or network), computer load, data precision, data record error, understanding divergence of failure definition, etc. Many software reliability growth models have been developed over the years. Within these models, one can distinguish two main categories as predictive models and assessment models. Predictive models address the reliability of software early in the life cycle at the requirements or at the primary design level or at the detailed design level in the waterfall life cycle process or in the first spiral of spiral software development process. Predictive models can be used to assess the risk of developing software under a given set of requirements and for specified personnel before the project truly starts. Assessment models evaluate present and future software reliability from failure data gathered when the integration of software starts.

In an empirical software reliability model, a relationship or a set of relationships between software reliability measures and appropriately defined software metrics are developed using empirical results available from the past data. This model can then be applied to measure software reliability for which one can have the required software metrics. The major issues of this modeling technique are identification of appropriate software metrics and development of right type and form of relationships between these metrics and the reliability measures. The analytical modeling of software reliability comprises of four steps. In step 1, the assumptions associated with the software test procedure are defined. The test procedure is developed in step 2 based upon the assumptions In step 3, the parameters for the model using the collected data are obtained. The performance prediction is done in Step 4. Two major types of analytical models are dynamic models and statistic models. In dynamic software reliability model, the time dependent behavior of software failures is captured. In a statistical model, no reference is made to the time dependent behavior of software failures. A number of analytical models have been proposed to address the problem of software reliability measurement (1). These approaches are based mainly on the failure history of software and can be classified according to the nature of failure process studied as times between failures models, failure count models, fault seeding models, input domain based models.

Yanyan et.al. [2] use the Goel Okumoto model as the non homogeneous Poisson process and double exponential smoothing as time series analysis composing stochastic process model. Deghuamei [3] proposed a grey model for software system reliability estimation. In this model the forecast data are obtained by grey model for software system reliability estimation. Yong Cao and Qing-Xin Zhu [11] identified the fractal relationship between cumulative time of Software failure and accumulative number of software faults. Also the fractals method could forecast the next software failure time. Khaled M.S. Faqih [4] explore critical factors and issues that impede the performance of software reliability modeling. Chen Zhongmin, Wu Yeqing [9] considered the failure data in testing phase as a time series and carried out the forecast on the basis of stochastic time series model. Khalaf Khatatneh [5] proposed a growth reliability model using fuzzy logic technique and applied on a custom set of test data. N.Rajkiran and V.Ravi [12] used a nonlinear ensemble trained using backpropagation neural network in predicting software reliability forecasting. Sultan H. Aljahdali and Mohammed E.El-Telbany[14] measured the predictability of software reliability using ensemble of models trained using genetic algorithms. The prediction of software faults during the testing process is done throught he historical faults data. Ra Kiran et.al., [6] developed ensemble models to accurately forecast software reliability efficiently. Various statistical and intelligent techniques constitute the ensembles. They are

multiple linear regression splines, back propagation neural network, dynamic evolving neuro fuzzy interface system and tree net. Yogesh Singh et.al., [7] used prediction of fault prone software modules using statistical and machine learning methods. Liviu Adrian Cotfas and Andreea Diosteanu [15] proposed a methodology for evaluating and assuring reliability in semantic web service composition applications based on the concept of abstract web services. Jung-Hua Lo [10] proposed Support Vector Machines based model for software reliability forecasting model. The parameters of support vector machine are determined by genetic algorithm. Ryad Zemouri and Paul Ciprian Patic [13] proposed an online adaptive reliability prediction model using evolutionary connectionist approach based on multiple-delayed input single output architecture. Through the software failure time data, a fuzzy min-max algorithm is used to optimize the k-Gaussian nodes and then determination and initialization of K-Centers of the neural network was done.

2. RELIABILITY MODEL FOR AUTOREGRESSIVE PROCESS

Time series analysis methods are suitable for forecasting sequential events, when successive observations are autocorrelated. The current study focuses on time series prediction of software fault occurrences. There are several statistical methods that yield high predictive power. The application of these methods requires expertise in parameter adjustment and also the methods are data intensive. Hence, a simple model that is less expensive in terms of computation and data requirement is advisable for prediction. One such technique that has been successfully used in software effort prediction is ARIMA. George E.P.Box (2003) et.al (8), proposed the ARIMA modeling method for forecasting based on historical data. The general equation of ARIMA (p,d,q) is

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots\ldots\ldots\ldots + \phi_P Y_{t-P} + \varepsilon_t - \omega_1 \varepsilon_{t-1} - \omega_2 \varepsilon_{t-2} - \ldots\ldots\ldots\ldots\ldots - \omega_q \varepsilon_{t-q}.$$

Here, Y_t is the time series of variable Y and ϕ_t is the coefficient associated with Y_t , which is to be estimated using the method of least squares. ε_t is the defect term, assumed to be independent , identically distributed variables sampled from a normal distribution with mean zero. ω_t is the coefficient associated with ε_t . This model is parameterized as ARIMA (p,d,q) where p is the order of autoregressive component , d is the order of differenced component and q is the order of moving average component. For example the model described as ARIMA(0,1,2) means that it contains 0 autoregressive (p) parameters and 2 moving average (q) parameters which were computed for the series after it was differenced once. The four steps in ARIMA modeling strategy followed in this study comprises of identification, estimation, diagnostic testing and application.

3. RELIABILITY MODEL FOR BIVARIATE AUTOREGRESSIVE (BAR) PROCESS:

A bivariate process $\left\{ \begin{pmatrix} X_t \\ Y_t \end{pmatrix}, t \in T \right\}$ is said to follow a bivariate autoregressive process of order 1 (BAR(1)) if it can be expressed as $Z_t = \mu + \Phi\left(Z_{t-1} - \mu\right) + \in_t$,

$t = 1, 2, 3, \ldots, T$;

$$where, \quad Z_t = \begin{pmatrix} X_t \\ Y_t \end{pmatrix}, \ \mu = \begin{bmatrix} \mu_X \\ \mu_Y \end{bmatrix},$$

$$\Phi = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} and \ \in_t = \begin{pmatrix} e_{X_t} \\ e_{Y_t} \end{pmatrix}$$

and $\in_t \ of \ Z_t$, follows a bivariate Gaussian Distribution whose mean vector is a null vector.

The Variance –Covariance matrix is expressed as :

$$\sum = \begin{pmatrix} \sigma_{e_{X_t}}^{\ 2} & \rho\sigma_{e_{X_t}}\sigma_{e_{Y_t}} \\ \rho\sigma_{e_{X_t}}\sigma_{e_{Y_t}} & \sigma_{e_{Y_t}}^{\ 2} \end{pmatrix}$$

The probability density function of $\in_t$ is

$$f\left(e_{X_t}, e_{Y_t}\right) = \left(\frac{1}{2\pi\,\sigma_{e_{X_t}}\,\sigma_{e_{Y_t}}\sqrt{1-\rho^2}} \right) * \exp\left\{ -\frac{1}{2\left(1-\rho^2\right)} \left(\frac{e_{X_t}^2}{\sigma_{e_{X_t}}^2} - \frac{2\rho e_{X_t} e_{Y_t}}{\sigma_{e_{X_t}}\sigma_{e_{Y_t}}} + \frac{e_{Y_t}^2}{\sigma_{e_{Y_t}}^2} \right) \right\}$$

$$-1 < \rho < 1, \ \sigma_{e_{X_t}} > 0 \ and \ \sigma_{e_{Y_t}} > 0$$

This process reduces to univariate AR(1) process if $\phi_{12}, \phi_{21} \ and \ \phi_{22}$ reduce to 0.

Since the residual terms in the sample parts of a time series starts from $u = 2$ in BAR(1) process, i.e. $e_{X_{t,u}} = X_{t,u} - \phi_{11} X_{t,u-1} - \phi_{12} Y_{t,u-1}$ and

$e_{Y_{t,u}} = X_{t,u} - \phi_{21} X_{t,u-1} - \phi_{22} Y_{t,u-1}$.

$$where, \ \Phi = \left\{ \phi_{11}, \phi_{12}, \phi_{21}, \phi_{22} \right\}$$

be the set of model parameters and n is the number of observations in the series. (George et.al., (2003)). These model parameters are estimated using the ordinary least squares estimate of Φ_t as:

$$\hat{\Phi}_t = \left(X_t^T X_t\right)^{-1} X_t^T \Upsilon_t \qquad \text{for } t = 1, 2, \ldots, N$$

where, the BAR(1) is written in the format as :

$$\begin{bmatrix} X_{t,2} & Y_{t,2} \\ X_{t,3} & Y_{t,3} \\ . & . \\ . & . \\ . & . \\ X_{t,n} & Y_{t,n} \end{bmatrix} = \begin{bmatrix} X_{t,1} & Y_{t,1} \\ X_{t,2} & Y_{t,2} \\ . & . \\ . & . \\ . & . \\ X_{t,n-1} & Y_{t,n-1} \end{bmatrix} * \begin{bmatrix} \phi_{11t} & \phi_{21t} \\ \phi_{12t} & \phi_{22t} \end{bmatrix} + \begin{bmatrix} \in_{X_{t,2}} & \in_{Y_{t,2}} \\ \in_{X_{t,3}} & \in_{Y_{t,3}} \\ . & . \\ . & . \\ . & . \\ \in_{X_{t,n-1}} & \in_{Y_{t,n-1}} \end{bmatrix}$$

This can be represented as $\Upsilon_t = X_t \Phi_t + \xi_t$

Where

$$\Upsilon_t = \begin{bmatrix} X_{t,2} & Y_{t,2} \\ X_{t,3} & Y_{t,3} \\ . & . \\ . & . \\ . & . \\ X_{t,n} & Y_{t,n} \end{bmatrix}, \; X_t = \begin{bmatrix} X_{t,1} & Y_{t,1} \\ X_{t,2} & Y_{t,2} \\ . & . \\ . & . \\ . & . \\ X_{t,n-1} & Y_{t,n-1} \end{bmatrix}, \; \Phi_t = \begin{bmatrix} \phi_{11t} & \phi_{21t} \\ \phi_{12t} & \phi_{22t} \end{bmatrix} \text{ and } \xi_t = \begin{bmatrix} \in_{X_{t,2}} & \in_{Y_{t,2}} \\ \in_{X_{t,3}} & \in_{Y_{t,3}} \\ . & . \\ . & . \\ . & . \\ \in_{X_{t,n-1}} & \in_{Y_{t,n-1}} \end{bmatrix}$$

4. EXPERIMENTAL RESULTS

Table 1: Faults Occurred, Estimated faults of BAR(1), AR(1), AR(2), ARIMA (1,1,1)

		BAR(1)	AR(1)	AR(2)	ARIMA
	Observed	Estimated	Estimated	Estimated	Estimated
Day	Faults	Faults	Faults	Faults	Faults
0	4	3.78	4.86	4.61	3.88
5	4	3.78	5.09	4.84	4.04
10	4	3.54	3.5	3.15	3.25
15	2	2.01	4.03	3.78	2.64
20	2	1.77	4.03	3.57	2.43
25	5	4.43	5.63	4.61	3.89
30	2	1.77	6.69	7.17	4.55
35	5	4.43	8.29	8	6.24
40	3	2.66	4.56	5.49	5.71
45	3	2.66	4.56	4.63	4.38
50	4	3.54	5.63	4.61	3.86
55	11	9.74	6.69	6.74	8
60	12	10.63	8.82	8.42	9.91
65	9	7.97	7.22	7.16	10.06
70	13	11.51	9.88	9.26	10.78
75	4	3.54	4.56	4.2	6.59
80	3	2.66	4.03	6.14	5.38
85	2	1.77	2.97	2.73	3.63
90	11	9.74	2.97	2.94	5.99
95	1	0.89	4.56	3.99	2.42
100	1	0.89	3.5	2.93	1.7
105	1	0.89	3.5	2.93	1.28

Figure 1 : Time Series Plot of Faults

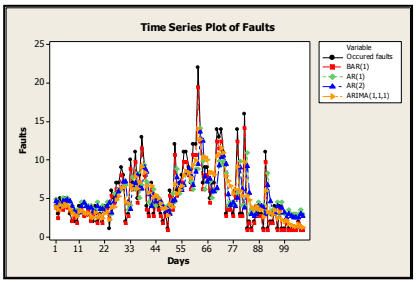
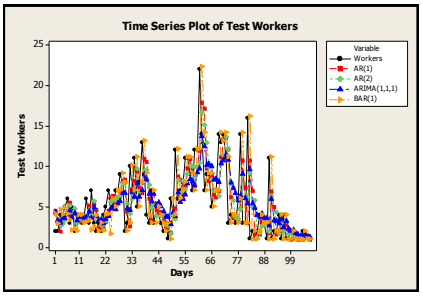


Table 2: Test Workers, Estimated Workers of BAR(1), AR(1), AR(2), ARIMA (1,1,1)

	Observed	BAR(1)	AR(1)	AR(2)	ARIMA
	Test	Estimated	Estimated	Estimated	Estimated
Day`	workers	Test workers	Test workers	Test workers	Test Workers
0	5	4.19	5.29	4.92	4.31
5	5	4.19	4.69	4.52	4.04
10	4	4.07	3.67	3.91	3.77
15	3	2.15	5.71	5.33	3.65
20	2	2.03	4.18	3.7	2.95
25	5	5.09	5.71	5.95	4.7
30	2	2.03	6.72	7.18	4.79
35	5	5.09	8.25	7.99	6.28
40	3	3.05	4.69	5.55	5.75
45	3	3.05	4.69	4.72	4.46
50	4	4.07	5.71	4.71	3.9
55	11	11.19	6.72	6.76	7.89
60	12	12.21	8.76	8.39	9.78
65	9	9.16	7.23	7.17	10.02
70	13	13.23	9.78	9.21	10.73
75	4	4.07	4.69	4.31	6.7
80	3	3.05	4.18	6.17	5.46
85	2	2.03	3.16	2.88	3.75
90	11	11.19	3.16	3.09	5.98
95	1	1.02	4.69	4.11	2.46
100	1	1.02	3.67	3.08	1.75
105	1	1.02	3.67	3.08	1.31

Figure 2: Time Series Plot of Test Workers



AR (1) model : $X_t = 2.4345 + 0.532\, X_{t-1}$

AR (2) model : $X_t = 1.8755 + 0.4203\, X_{t-1} + 0.2148\, X_{t-2}$

ARIMA (1,1,1) model : $X_t = -0.0208 + 0.1705\, X_{t-1}$

BAR (1) Model :

$$\begin{pmatrix} X_t \\ Y_t \end{pmatrix} = \begin{pmatrix} X_{t-1} \\ Y_{t-1} \end{pmatrix} * \begin{pmatrix} 0.6482 & 0.9014 \\ 0.2373 & 0.1160 \end{pmatrix}$$

The above AR(1) model, AR(2) Model, ARIMA (1,1,1) model and BAR(1) model are constructed by using the Observed faults and Observed test workers data presented in table 1, table 2 respectively. Also, the time series plot for test workers and estimated test workers was plotted for AR(1) model, AR(2) Model, ARIMA (1,1,1) model and BAR(1) models against the days and presented in figure 1. Figure 2 gives the time series plot for test workers and estimated test workers of the AR(1) model, AR(2) model, ARIMA (1,1,1) model and BAR(1) model.

5. CONCLUSIONS AND FUTURE WORK

The proposed bivariate autoregressive model of order 1 uses the past number of faults together with the test workers to build a model structure that can provide an estimate of future faults. This approach attempts to model the relationship between measured faults and previous faults in a recurrent relation. The recurrent relation is then used to provide an approximate new measurement of the future faults. A comparison between AR(1) model, AR(2) model, ARIMA(1,1,1) was also provided. The results of proposed BAR(1) were promising than the existing univariate AR models. This proposed BAR(1) model can be extended to Multivariate AR model which is our next goal**.**

6. REFERENCES

[1] Jelinski,Z., and Moranda,P., 1972, Software Reliability Research, In Statistical Computer Performance Evaluation. 465-484.

[2] Yanyan Zheng, RenZuo Xu, 2008, A Composite Stochastic Process model for Software Reliability, International Conference on Computer Science and Software Engineering, 658 – 661.

[3] Denghua Mei, 2007, Novel Model for Software Reliability by Grey System Theory, 513-516.

[4] Khaled M.S. Faqih, 2009, What is Hampering the performance of Software Reliability Models? A Literature Review, Proceedings of the International Multi Conference of Engineers and Computer Scientists.

[5] Khalaf Khatatneh, 2009, Software Reliability Modeling using Soft Computing Technique, European Journal of Scientific research, 154-160.

[6] Raj Kiran,N., and Ravi,V., 2007, Software Reliability Prediction by Soft Computing Techniques, The Journal of Systems and Software.

[7] Yogesh Singh, Arvinder Kaur, Ruchika Malhotra, 2010, Prediction of fault prone Software modules using Statistical and Machine learning methods, International Journal of Computer Applications, 6-13.

[8] George E.P.Box, Gwilym M. Jenkins, and Gregpry C. Reinsel, 2003, Time Series Analysis, Forecasting and Control.

[9] Chen Zhongmin, Wu Yeqing, 2010, The application of theory and method of time series in the modeling of software reliability, Proceedings of International Conference on Information technology and Computer Science, 340-343.

[10] Jung-Hua Lo, 2010, Predicting Software Reliability with Support Vector Machines, International Conference on Computer Research and Development, 765-769.

[11] Yong Cao and Qing-Xin Zhu, 2010, The Software reliability forecasting Method using Fractals, Information technology Journal , 331-336.

[12] N.Raj Kiran, V.Ravi, 2007, Software reliability Prediction by Softcomputing techniques, The Journal of Systems and Software.

[13] Ryad Zemouri, Paul Ciprian Patic, 2010, Recurrent Radial Basis function Network for failure time series prediction, World academy of Science, Engineering and technology, 748 – 752.

[14] Sultan H. Aljahdali and Mohammed E.El-Telbany, 2008, Genetic Algorithms for Optimizing Ensemble of Models in Software reliability Prediction , ICGST-AIML Journal, 5-13.

[15] Liviu Adrian Cotfas and Andreea Diosteanu, 2010, Software Reliability in Semantic Web Service Composition Applications, Informatica Economica, Vol 14, No 4, 48 – 56.