JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 6, September 2009

Software Forecasting Models

Norman Schneidewind*
Naval Postgraduate School, Pebble Beach, CA 93953

DOI: 10.2514/1.38053

Several forecasting models were evaluated with respect to their forecasting accuracy using NASA Space Shuttle failure data. Simple models like exponential smoothing are attractive if the forecasting range can be limited. This is the case because these models use the most recent data in making a forecast. In other cases, where forecasts must be made over a wide range, software reliability models are superior. The reason is that these models use a larger set of data in making parameter estimates. This feature results in more accurate forecasts over a wider range of time periods in the future. We found that significant improvements could be made to forecasting accuracy for all models by the simple process of modifying original forecasts based on the relative errors of those forecasts.

I. Introduction

OUR purpose is to experiment with various forecasting models to see how accurate they are for forecasting NASA Space Shuttle software reliability metrics. Shuttle failure data is used because we have the data, but the experimental procedure could be applied to any application. One category of model is software reliability models that are included in IEEE/AIAA standards [1]. It is usually assumed that these models are the most appropriate for forecasting software reliability. However, this might not always be the case. For example, some researchers found that applying simple exponential smoothing to the time series resulted in higher accuracy for forecasts than more sophisticated models. This leads them to believe that in certain cases, depending on the characteristics of the time series, naïve methods of forecasting may produce more accurate results [2]. Therefore, we experimented with several software reliability models and with simpler models—exponential smoothing, autoregressive integrated moving average, moving average, and regression. Plots were made to compare the forecasts of reliability metrics against the actual data. The mean relative error (MRE) and mean square error (MSE) were used to compare forecasting accuracy [3].

A. Software Reliability Growth Modeling

Reliability growth for software is the improvement of software reliability over time, accomplished through the systematic removal of software faults. The rate at which reliability grows depends on how fast faults can be uncovered and removed. A software reliability growth model allows project management to track the progress of the software's reliability through a series of tests.

If the assessed growth falls short of the planned growth, management will have sufficient notice to develop new strategies, such as the reassignment of resources to attack identified problem areas, adjustment of the project time frame, and reexamination of the feasibility or validity of requirements.

Measuring and projecting software reliability growth requires the use of an appropriate software reliability model that describes the variation of software reliability with time. The parameters of the models are estimated from failure data collected during test [4].

Received 13 April 2008; accepted for publication 20 June 2009.
* Fellow of the IEEE, IEEE Congressional Fellow, US Senate 2005, ieeelife@yahoo.com

Thus, given the importance of software reliability modeling, as described above, we want to ensure that forecasts are accurate and, if the models do not forecast accurately, to consider and evaluate alternate models.

II. Software Reliability Forecasting Models

A. Brooks and Motley Binomial Model

Several software reliability models were used in the forecasting experiment to see how they would fare relative to other models, such as exponential smoothing and regression. Among these models is the Brooks and Motley binomial model (BMBM) that uses failure count data for parameter estimation. This model is among those that recognize that not all the code is tested equally and that the code may not be complete at the time of testing. The model factors into the forecasting equations of the modules that are under test [5]. The cumulative number of failures $F(T)$ forecast in test period T is given by

$$F(T) = (1 - (1 - q)^{E_T}) \tag{1}$$

where q is the constant probability of failure detection in each period and E_T is the amount of test effort expended in period T [6].

B. Yamada S Shaped Model

A second software reliability model used in the experiments is the Yamada S Shaped Model (YSSM) that derives its name from assuming that a "learning curve" scenario applies (i.e., the number of faults found slowly ramps up as the testers become familiar with the code, accelerates up quickly, and then levels off as the remaining faults become more difficult to find) [7]. This model's forecasted number of cumulative failures $F(T)$ is given by

$$F(T) = a(1 - (1 + bT)e^{-bt}) \quad \text{with both } a, b > 0 \tag{2}$$

For given values of $F(T)$, Eqs. (1) and (2) can be solved for T, the time forecasted to detect $F(T)$ cumulative failures during test or operation.

Time to next failure T, cumulative failures $F(T)$, and mean number of failures in interval T, $m(T)$, are forecast for BMBM and YSSM by using the SMERFS software reliability modeling tool [8].

C. Schneidewind Software Reliability Model

The third software reliability model used in the experiment is the Schneidewind software reliability model (SSRM) that has the feature of only using the most relevant failure data in parameter estimates [9]. This feature is implemented by using the collected failure data to find the value of the parameters s, α, and β that maximize the likelihood function. These parameter values are used in forecasting equations (3), (4), (4a), and (7).

To forecast *time to next* F_t *failure*(s), when the current time is t and $X_{s,t}$ failures in the range s, t have been observed, Eq. (3) is used.

$$T(t) = \log\left[\frac{\alpha}{(\alpha - \beta(F_t + X_{s,t}))/\beta}\right] - (t - s + 1) \quad \text{for } \alpha > \beta(F_t + X_{s,t}) \tag{3}$$

Time to next failure is a critical forecast for safety critical systems such as the Shuttle. It is desired that it exceed the mission duration [10]. It is also desirable that the differences in *time to next failure* forecasts [11] *increase*, reflecting increasing mission safety, as shown in (3a).

$$\Delta T(t, 1) < \Delta T(t, 2), \ldots, \Delta T(t, n, -1) < \Delta T(t, n) \tag{3a}$$

where n is the number of tests.

The forecasted *cumulative number of failures* $F(T)$ to occur at time T is given in equation (4).

$$F(T) = \frac{\alpha}{\beta}[1 - e^{-\beta(T-s+1)}] + X_{s-1} \tag{4}$$

The *mean number of failures in the interval* $T, T+1$ is forecasted in equation (4a):

$$m(T) = \frac{\alpha}{\beta}[e^{-\beta(T-s+1)} - e^{-\beta(T-s+2)}] \tag{4a}$$

Since $R(t)$, the reliability in interval T, is equal to the probability of $x_T = 0$ failures in the Poisson process, as in equation (5):

$$P(x) = (m(T)_T^x e^{-m(T)})/X_T! \tag{5}$$

As a consequence of Eq. (5), we derive Eq. (6):

$$R(T) = P(0) = e^{-m(T)} \tag{6}$$

Then substituting equation (4a) into equation (6), we arrive at the reliability forecasting equation (7):

$$R(T) = e^{-[(\alpha/\beta)\,[e^{-\beta(T-s+1)} - e^{-\beta(T-s+2)}]]} \tag{7}$$

In order to compute the reliability forecasting error, it is necessary to compute the actual reliability at time T in equation (7a), where x_T is the number of observed failures in the interval T and F is the total number of observed failures.

$$R_a(T) = 1 - (x_T/F) \tag{7a}$$

Furthermore, denoting R_s as the specified reliability, then the predicted reliability $R(T)$ should obey (7b) for all values of T as follows:

$$R(T) \geqslant R_s \tag{7b}$$

III. Exponential Smoothing Model

The rationale for using exponential smooting model (ESM) to forecast future values in a time series is that, according to [12], significant changes are possible in the data. Therefore, some provision must be made for easily adjusting the number of past observations that are incorporated in the forecast. Whenever there is a significant change in the pattern, computations should be made using only the most recent observations. We can apply ESM—a simple, yet effective model for forecasting the occurrence of events at time T—based on the actual the value at time $T-1$ and the forecast at time $T-1$ [12,13].

Traditionally, ESM forecasts are made by guessing at the value of the smoothing parameter α. While guessing may yield an accurate forecast, we think this is an unsatisfactory procedure for two reasons: 1) it is difficult to know which value of α to select and 2) a constant α does not account for variations in the data. Therefore we develop a method below of estimating a variable α from known data.

Definitions

1. Item: x
2. α: smoothing parameter
3. Actual x at time $T = x_T$
4. Actual x at time $(T-1) = x_{T-1}$
5. Actual x at time $(T-2) = x_{T-2}$
6. Forecast at time $(T-1) = x_{(T-1)}$

Using definitions 1, 2, 5, and 6, we develop Eq. (8) as follows:

$$\text{Forecast of } x \text{ at time } (T-1) = x(T-1) = \alpha x_{T-2} + (1-\alpha)x(T-2) \tag{8}$$

Then using definitions 1, 2, 4, and Eq. (8) recursively, we produce Eq. (9) as follows:

$$\text{Forecast of } x \text{ at time } T = x(T) = \alpha x_{T-1} + (1-\alpha)x(T-1) \tag{9}$$

Expand the forecast of x at time T, using Eqs. (8) and (9) to produce equation (10) as follows:

$$T = x(T) = \alpha x_{T-1} + (1-\alpha)x(T-1) = \alpha x_{T-1} + (1-\alpha)*(\alpha x_{T-2} + (1-\alpha)x(T-2)) \tag{10}$$

In order to not guess at a *constant* value of α, which is the traditional approach, we develop the following equations for estimating α, corresponding to the three possibilities for the relationship between x_{T-1} and x_{T-2}. We use a variable value of α. This value cannot be estimated at time T because as shown in equations (11), (12), and (13), it depends on prior values of the data. Therefore, from this point forward, we use α_{T-1} to indicate this dependency.

$$\textit{If } x_{T-1} > x_{T-2}, \text{ then } \alpha_{T-1} = (x_{T-1} - x_{T-2}/x_{T-1} \quad (\text{for } x_{T-1} > 0) \tag{11}$$

$$\textit{If } x_{T-1} < x_{T-2}, \text{ then } \alpha_{T-1} = (x_{T-2} - x_{T-1})/x_{T-2} \quad (\text{for } x_{T-2} > 0) \tag{12}$$

$$\textit{If } x_{T-1} = x_{T-2}, \text{ then } \alpha_{T-1} = 0 \tag{13}$$

Then, setting the forecast at time $(T-2)$ = the actual value: $x(T-2) = x_{T-2}$, we generate equation (14). Note that this is necessary because, of course, we must use the starting value x_{T-2} for this recursive process.

$$\text{Forecast of } x \text{ at time } T = x(T) = \alpha_{T-1}x_{T-1} + ((1-\alpha_{T-1}) * \alpha_{T-1}x_{T-2}) + ((1-\alpha_{T-1})^2 x_{T-2}) \tag{14}$$

Once the appropriate value of α_{T-1} is estimated, using equation (11), (12), or (13), and Eq. (10) is simplified, the forecast of x at time T is made in Eq. (15) as follows:

$$x(T) = \alpha_{T-1}(x_{T-1} - x_{T-2}) + x_{T-2} \tag{15}$$

As was the case for SSRM, it is desirable for certain time series, such as *time to next failure*, to be increasing as the number of tests n increases. This is expressed in equation (15a).

$$\Delta x(T) < \Delta x(T-1), \ldots, \Delta xT(T-n-1) < \Delta xT(T-n) \tag{15a}$$

A. Geometric ESM

A variant of ESM is to use an α that increases at an increasing rate: the geometric ESM (GESM) [14]. This concept is implemented by using e^{α} in the forecasting equations. In our experiments, this approach has provided better forecasting accuracy than ESM.

IV. Moving Average Model

The rationale for moving average model (MAM) is that since data is usually varying across time periods, it makes sense to capture the variation by computing a varying average, rather than a single average across all time periods. This idea is expressed in equation (16).

$$x(T) = \frac{\sum_{i=1}^{n} x_{T-i}}{n} \tag{16}$$

where $x(T)$ is the forecast for time T of item x, x_{T-i} is *actual* value of item x for ith period preceding T, and n is the number of time periods to include in the moving average.

When MAM is used to forecast differences in $x(T)$ that should be an increasing series (i.e., *time to next failure*), Eq. (15a) comes into play.

V. Regression Model

Regression involves making the "best" fit of a dependent variable to an independent variable (e.g., fit cumulative failures to software test time), based on a fit criterion such as least squares. Unfortunately, if major jumps occur in the *future* data that were not included in the fitted data, the regression equation could result in inaccurate forecasts in the future space, even though there was a good fit in the historical space. We will illustrate this phenomenon in one of our forecasting experiments.

VI. Simple ARIMA Model

A simple version of the autoregressive integrated moving average (ARIMA) model uses the immediate past value x_{T-1} and the average of the *difference* of two immediate past values x_{T-1} and x_{T-2}, to forecast the current value $x(T)$ [15] as follows:

$$x(T) = x_{T-1} + \frac{x_{T-1} - x_{T-2}}{2} \tag{17}$$

Again, it may be necessary to forecast differences in $x(T)$ when it is desirable that the time series increase. As in the case of GESM and MAM, this can be accommodated by employing Eq. (15a).

VII. Failure Data Used in Experiments

The failure data used in the experiments is shown in the Appendix: NASA Space Shuttle release OI4. The first seven entries are the data used to estimate parameters for the software reliability models SSRM, YSSM, BMBM, and the regression model. The last six entries are compared with forecasted values in this range. In the case of GESM, ARIMA, and MAM, forecasts are made for only the eighth failure count interval due to the limitation of these models to forecast beyond the *actual* data.

VIII. Forecast Results

A. Time to Next Failure

Figure 1 displays the results for the three software reliability models, with BMBM registering as the most accurate, although forecasting accuracy, as computed by MRE, is not notable. This figure also shows that all the models forecast *time to next failure* to exceed the mission duration that is specified as a typical Shuttle mission duration.

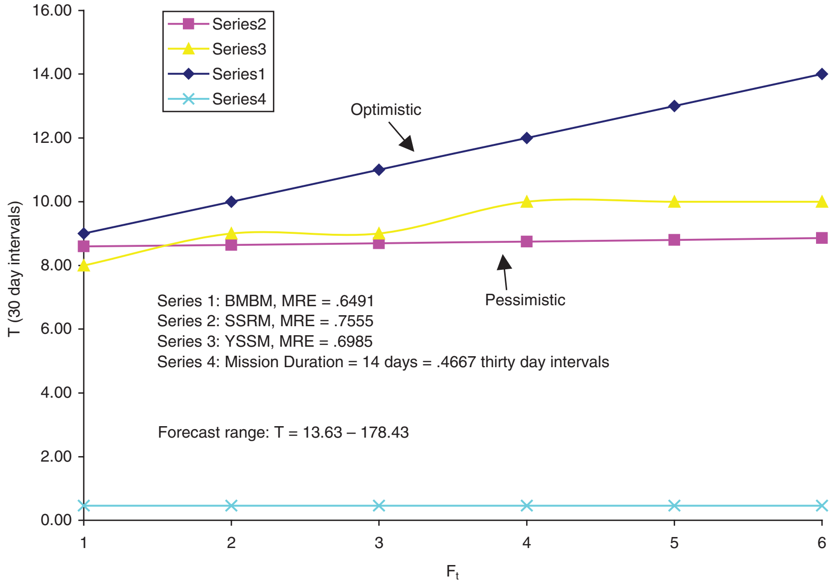

Fig. 1 NASA Space Shuttle OI4: predicted time to next failure T vs number of failures F_t.

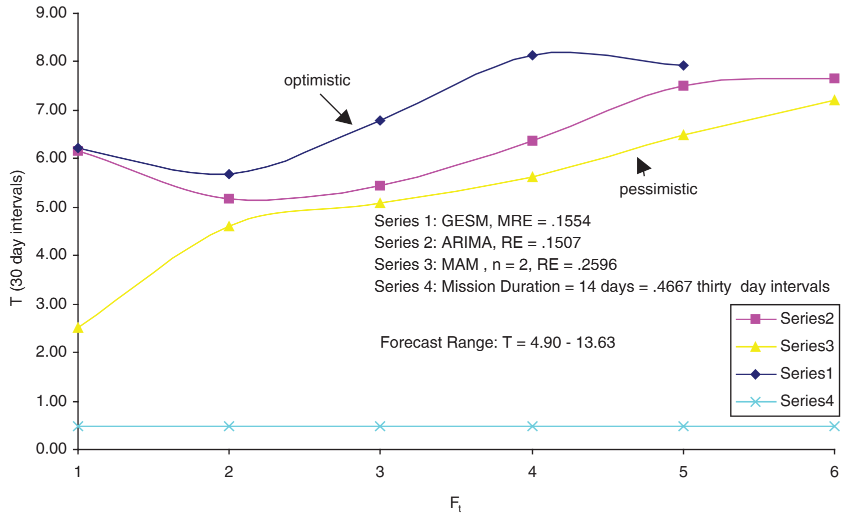

Fig. 2 NASA Space Shuttle OI4: forecast time to next failure t vs number of failures F_t.

It is advantageous to forecast using more than one model so that bounds are provided between the pessimistic and optimistic forecasts.

Figure 2 shows GESM, ARIMA, and MAM, $n = 2$, with all models providing better forecasting accuracy than the software reliability models in Fig. 1. However, we hasten to add that the models in Fig. 2 have limited forecasting range. These results suggest that software reliability models should be used when it is desired to forecast reliability over an extended range, whereas GSEM, ARIMA, and MAM would be the choice if the interest is to forecast for just *one* period beyond the end of the failure data.

Figure 3 shows what is perhaps a counterintuitive picture of the moving average computed for $n = 2, 3, 4$, and 5 samples, because the MRE of the forecasts is *increasing* rather than *decreasing*. The reason for this is that the *time to next failure* series is monotonically increasing, and the average of n previous samples, can never catch up with the increasing actual data.

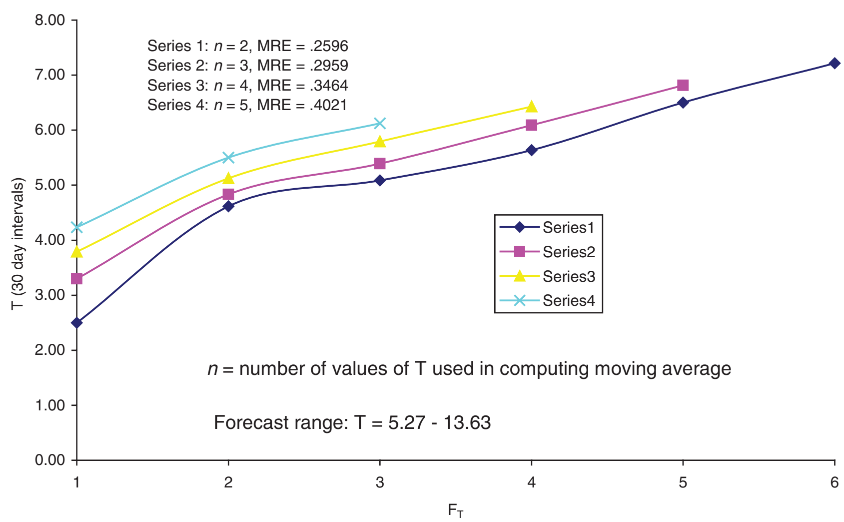

Fig. 3 NASA Space Shuttle OI4: moving average of time to next failure T vs number of failures F_t.

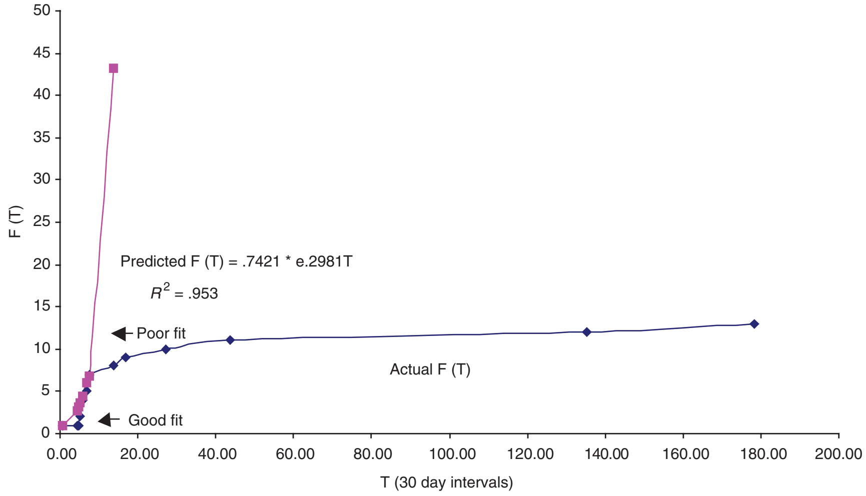

Fig. 4 NASA Space Shuttle OI4: cumulative failures $F(T)$ vs test time T.

A procedure for choosing the best value of n is to start with $n = 2$ and compute MRE. Then increase n. As soon as MRE *increases*, choose the previous value of n.

B. Cumulative Failures

Figure 4 demonstrates the aforementioned problem with regression analysis, wherein in the test time range $T = 0.67–7.43$ thirty-day intervals, there is a good fit. However, when we attempt to use the regression equation for forecasting cumulative failures beyond the range where the regression equation was fitted, $T = 3.63-178.43$ intervals, the fit is poor. In order to protect against inaccurate forecasts, several models should be used and their forecast errors compared in order to eliminate any outlier forecasts.

Another interesting result is shown in Fig. 5 where the Brooks–Motley model comes in first in the forecasting accuracy race. However, in general, it is difficult to tell a priori which is the best model to use for a given application. Therefore users should experiment with several models, comparing forecasts with *actual future* data.

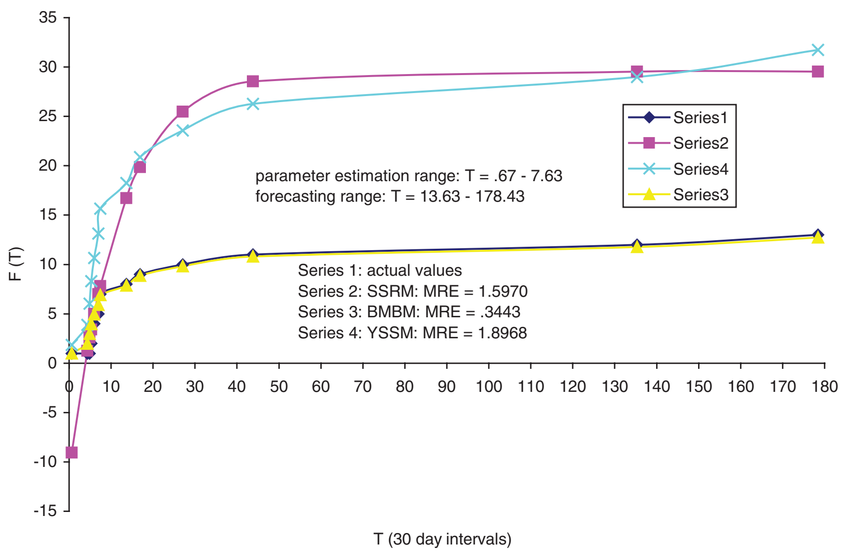

Fig. 5 NASA Space Shuttle OI4: cumulative failures $F(T)$ vs test time T.

Table 1 Forecasting model error analysis

Model	Number of forecasts for T	Number of forecasts for $F(T)$	T: time to next failure failure MRE	T: time to next next failure failure MSE	$F(T)$: cumulative failures MRE	$F(T)$: cumulative failures MSE
GESM	5	5	0.1554	6.81	**0.1708**	1.0822
ARIMA	6	6	**0.1507**	**6.35**	0.1815	0.9167
SSRM	6	13	0.7555	7926	1.5970	143
YSSM	6	13	0.6985	7607	1.8968	120
BMBM	6	13	0.6491	7224	0.3343	0.82
MAM ($n = 2$)	6	6	0.2596	8.49	0.3554	**0.4167**
Regression		8			1.1888	156

IX. Error Analysis

Two types of error analysis are used to assess forecasting accuracy: 1) MRE to evaluate error relative to the actual values and MSE to evaluate the variance of the difference between actual and forecasted values. These are shown in Eqs. (18) and (19), respectively, where A_T and P_T are the actual and forecasted values at software test time T, respectively, and n is the number of samples (i.e., software tests). Table 1 tabulates MRE and MSE for each of the forecasting models, for *time to next failure* and *cumulative failures*.

$$\text{MRE} = \frac{\sum_{i=1}^{n} |(A_T - P_T)/P_T|}{n} \tag{18}$$

$$\text{MSE} = \frac{\sum_{i=1}^{n} (A_T - P_T)^2}{n} \tag{19}$$

In performing forecasting error analysis, it is important to observe the trend of relative errors across the software tests to see whether the errors stabilize. From the example in Fig. 6, we can see that the software reliability models have this property. If this were not the case, there would be concern that as tests progress, *time to next failure* goals would be unattainable. The lesson learned is that when choosing a forecasting model, the error trends, using historical data, should be evaluated in terms of MRE and RE trend. Based on these criteria, the selected model would have the best prospect of providing the most accurate forecasts in the *future*. In our example, BMBM would be that model for *time to failure* forecasts.

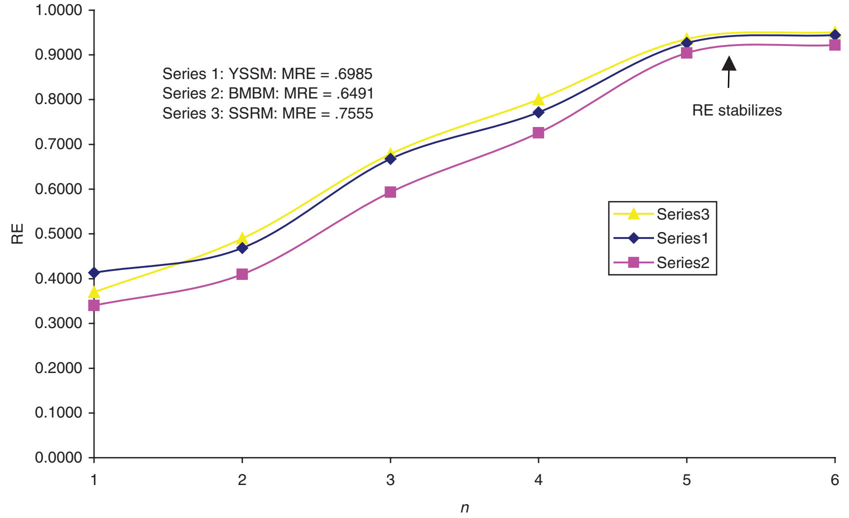

Fig. 6 NASA Space Shuttle OI4: forecasted time to next failure relative error RE vs number of tests n.

X. Forecasting Accuracy Improvement

Some researchers advocate selecting the best model by using models to forecast for a given set of failure data and then picking the one with the best accuracy [16,17]. They suggest comparing a forecast with the actual *future* observations, and recursively building up a sequence of such observations and forecast comparisons. From this sequence, we should be able to gain information about the accuracy of past forecasts, and make a decision about the current forecast (i.e., which model to trust, if any). The basic idea in analyzing model forecasting accuracy is to perform recursive comparison of forecasts with eventual actual data [18]. Another idea is to combine models such that one model that has the best accuracy in a lower range of the data is joined with another model that has the best accuracy in a higher range of the data [19]. This is an interesting idea but we did not find that it worked when applied to the data in the Appendix.

Our approach is to produce a modified forecast M_T in test time T by multiplying the relative error RE(T), computed in test time T, by the observed value A_{T-1} in test time $T-1$, and adding this result to the unmodified forecast P_T in test time T as follows:

$$M_T = P_T + (\text{RE}(T) * A_{T-1}) = P_T + \left[\frac{(A_T - P_T) * A_{T-1}}{A_T}\right] \tag{20}$$

Unlike the computation of relative error in Eq. (18), which is an absolute quantity, relative error in Eq. (20) is a signed quantity to allow the modified forecast M_T to reflect either a decrease or increase, depending on whether RE(T) is negative or positive, respectively.

A. Forecasting Results

1. *Time to Next Failure and Cumulative Failures*

We see that dramatic improvements in forecasting accuracy can be obtained can be obtained by applying Eq. (20) for both *time to next failure and cumulative failures*. For example, Fig. 7 shows the gains in forecasting accuracy achieved by YSSM and SSRM by modifying the time to next failure forecasts. This experience leads us to conclude that forecasts should be revised according to Eq. (20), even for reliability models like SSRM, YSSM, and BMBM, since it is a lot quicker and cheaper than the alternative of reestimating parameters and reforecasting, using more recent failure data (Table 2).

2. *Mean Number of Failures in Interval*

We continue to demonstrate the benefit of using Eq. (20) to improve forecasting accuracy by providing examples in Fig. 8 that show improvement for SSRM and ARIMA for mean number of failures in the interval T. The entire

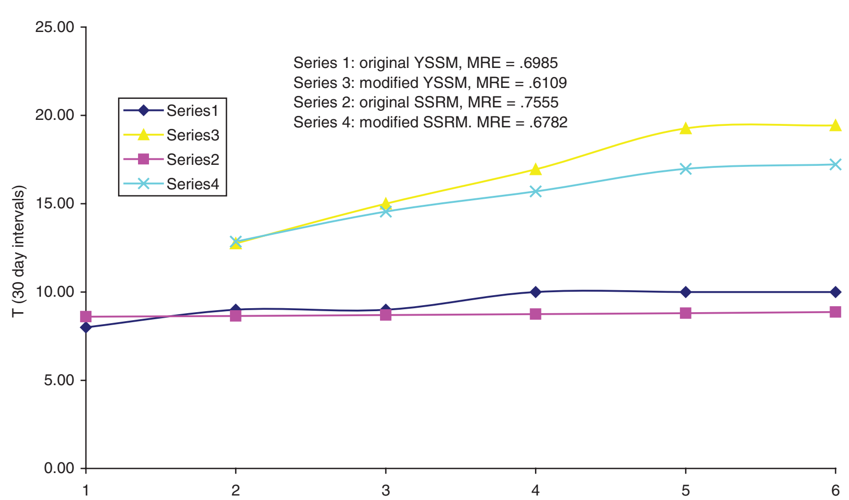

Fig. 7 NASA Space Shuttle OI4: forecasted time to next failure T vs number of failure F_t.

Table 2 Error analysis with forecasting model improvement: T and $F(T)$

Model	Number of forecasts for T	Number of forecasts for $F(T)$	Original forecast MRE	Improved forecast MRE	Original forecast MRE	Improved forecast MRE
GESM	5	5	0.1554	0.0889	0.1708	0.1464
ARIMA	6	6	**0.1507**	**0.0451**	**0.1815**	**0.0997**
SSRM	6	13	0.7555	0.6782	1.5970	0.9265
YSSM	6	13	0.6985	0.6109	1.8968	0.3236
BMBM	6	13	0.6491	0.2827	0.3343	0.0547
MAM ($n = 2$)	6	6	0.2596	0.0553	0.3554	0.0313
Regression		8			1.1888	0.1491

error record for all the models is shown in Table 3. Note that since the variable alpha GESM requires division by members of the data set to compute alpha, and there are zeros in the data set (see Appendix), it is infeasible to use GESM for these data.

3. Change in Time to Next Failure

As mentioned earlier, we hope to witness increasing values of *time to next failure* as the number of software tests increase, in order to have confidence that mission safety is increasing. In Fig. 9, MAM, $n = 2$, is confidence building but YSSM is not. However this is not the end of the story because, as shown in Fig. 9, MAM has a highly restricted forecast range compared with YSSM. Thus when evaluating forecast models, we should consider whether a desirable attribute has been achieved (e.g., increasing values of *time to next failure*) *and* whether the forecast range is adequate.

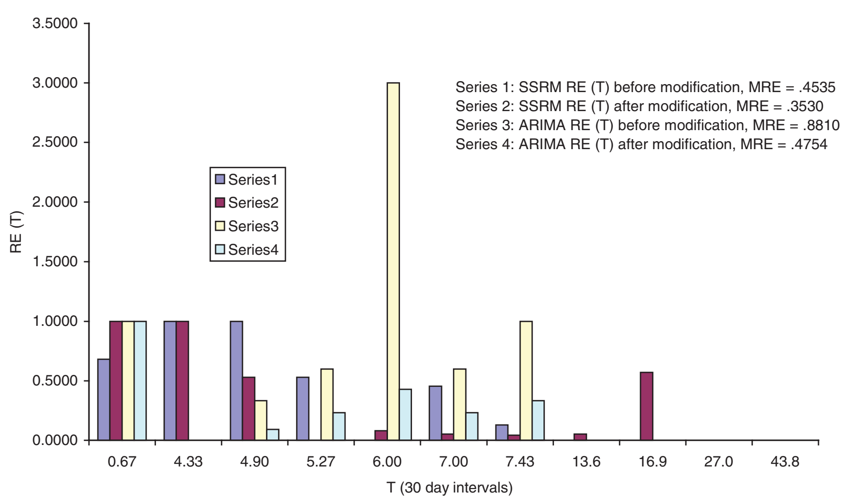

Fig. 8 NASA Space Shuttle OI4: forecasted mean failures in interval relative error RE(T) vs test time T.

Table 3 Error analysis with forecasting model improvement: $m(T)$

		Mean number of failures in interval (T): $m(T)$	
Model	Number of forecasts	MRE: original	MRE: modified
ARIMA	11	0.8810	0.4754
SSRM	9	0.4535	0.3530
YSSM	12	0.6064	0.4878
BMBM	9	**0.3568**	**0.1621**
MAM ($n = 2$)	11	0.7019	0.2994

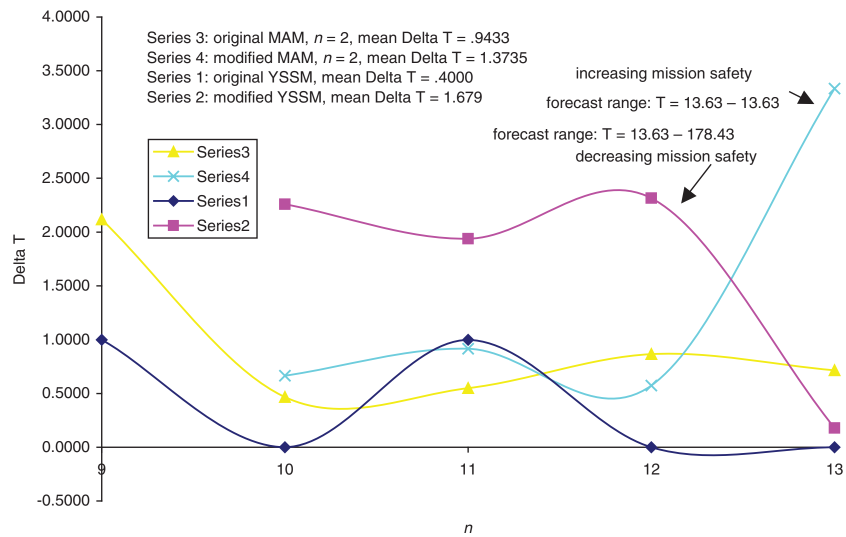

Fig. 9 NASA Space Shuttle OI4: delta *T* for time to next failure vs number of tests *n*.

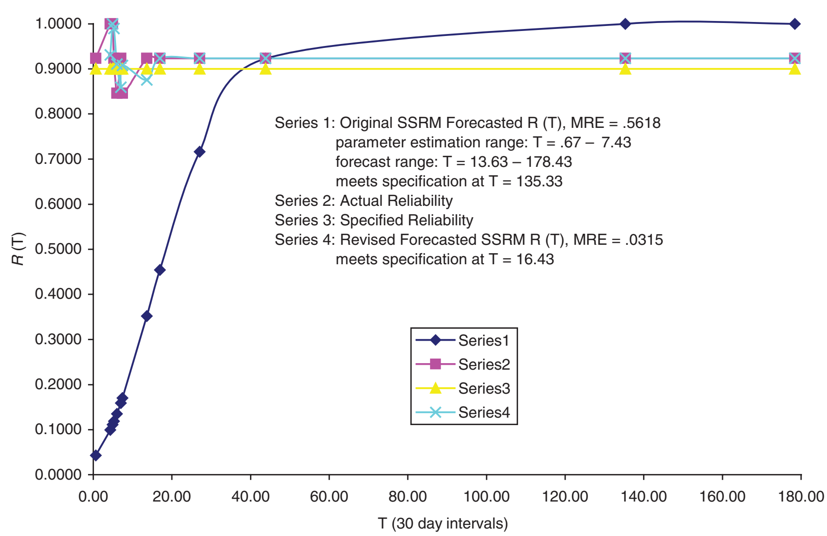

Fig. 10 NASA Space Shuttle OI4: reliability *R*(*T*) vs time *T*.

4. Reliability

Again, it is proven that making adjustments to original forecasts, using Eq. (20), are capable of significant improvement in reliability and reliability forecast error as evidenced by Fig. 10. In this figure we show that the original SSRM forecast does not meet the reliability specification until relatively late in the forecast range, whereas the revised forecast satisfies the specification much earlier.

XI. Summary of Forecasting Results

Based on Figs. 1–10 and Tables 1–3, for NASA Space Shuttle OI4, it is evident that there is no one model that is superior for all types of forecasts. When forecasts are required across a large range of software test time, software reliability models are appropriate. For a restricted range, GESM, ARIMA, and MAM are competitive with the software reliability models and, indeed, in some cases provide higher accuracy. For the latter models, they are limited in accurately forecasting beyond the range of the actual data.

We recognize that the applicability of the results may be limited to the Shuttle failure data although these data are typical of those for safety critical systems. Whether the results would be applicable to other software, such as commercial systems, could only be determined by additional research.

Appendix

NASA Space Shuttle Software Release O14

Thirty-day failure count intervals	Time to next failure (days)	Failure count	Cumulative failurers
0.67	20	1	1
4.33	130	0	1
4.90	147	0	1
5.27	158	1	2
6.00	180	2	4
7.00	210	1	5
7.43	223	2	7
13.63	409	1	8
16.93	508	1	9
27.07	812	1	10
43.80	1314	1	11
135.33	4060	1	12
178.43	5353	1	13

References

[1] IEEE/AIAA P1633™/Draft 14e, Recommended Practice on Software Reliability, March 2008.

[2] Abdollahian, M., and Foroughi, R., "Optimal Statistical Model for Forecasting Ozone," *Proceedings of the International Conference on Information Technology: Coding and Computing* (*ITCC'05*), Vol. I, 2005, pp. 169–173.

[3] Fenton, N. F., and Pfleeger, S. L., *Software Metrics: A Rigorous & Practical Approach*, 2nd ed., PWS Publishing Company, Boston, MA, 1997.

[4] Reliability Growth and Demonstration Testing, www.cs.colostate.edu/∼cs530/rh/section8.pdf

[5] Farr, W. *Handbook of Software Reliability Engineering*, edited by M. R. Lyu, IEEE Computer Society Press, Piscataway, NJ, and McGraw–Hill, New York, NY, 1996, pp. 92–93.

[6] Farr, W. H., "A Survey of Software Reliability Modeling and Estimating," Naval Surface Weapons Center, NSWC TR 82-171, September 1983.

[7] Xie, M., *Software Reliability Modelling*, World Scientific, Singapore, 1991.

[8] Farr, W. H., and Smith, O. D., "Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS) Users Guide," Naval Surface Warfare Center, NAVSWC TR-84-373, Revision 2, Dahlgren, VA.

[9] Schneidewind, N. F., "Reliability Modeling for Safety Critical Software," *IEEE Transactions on Reliability*, Vol. 46, No. 1, March 1997, pp. 88–98.
doi: 10.1109/24.589933

[10] Keller, T., and Schneidewind, N. F., "A Successful Application of Software Reliability Engineering for the NASA Space Shuttle," *Software Reliability Engineering Case Studies, Proceedings of the International Symposium on Software Reliability Engineering*, November 3–4, Albuquerque, NM, 1997, pp. 71–82.
doi: 10.1109/CSSRE.1997.637849

[11] Korzyk, A. D., Sr., "A Forecasting Model for Internet Security Attacks," Virginia Commonwealth University, J. G. VanDyke & Associates, http://csrc.nist.gov/nissc/1998/proceedings/paperD5.pdf

[12] Brown, R. G., *Smoothing, Forecasting and Prediction of Discrete Time Series*, Prentice–Hall, Upper Saddle River, NJ, 1963.

[13] Turner, W. C., Mize, J. H., and Nazemetz, J. W., *Introduction to Industrial and Systems Engineering*, 3rd ed., Prentice–Hall, Upper Saddle River, NJ, 1993.

[14] Decision 411 Course Outline 2005, http://www.duke.edu/∼rnau/411home.htm

[15] Introduction to ARIMA: Non-seasonal Models, http://www.duke.edu/∼rnau/411arim.htm

[16] Brocklehurst, S., Chan, P. Y., Littlewood, B., and Snell, J., “Recalibrating Software Reliability Models,” *IEEE Transactions on Software Engineering*, Vol. 16, No. 4, 1990.
doi: 10.1109/32.54297

[17] Lyu, M. R., and Nikora, A., “A Heuristic Approach for Software Reliability Prediction: The Equally-Weighted Linear Combination Model,” *Proceedings of the 2nd International Symposium on Software Reliability Engineering*, May 1991, pp. 172–181.
doi: 10.1109/ISSRE.1991.145376

[18] Qu, L., Chen, Y., and Yang, M., “A Dynamic Combination Forecast Model for Analysis Transport Volume Time Series,” *Proceedings of the Third International Conference on Natural Computation* (*ICNC 2007*), 2007, pp. 705–709.
doi: 10.1109/ICNC.2007.19

[19] Maia, A. L. S., de Carvalho, F. de A. T., and Ludermir, T. B., “Symbolic Interval Time Series Forecasting using a Hybrid Model,” *Proceedings of the Ninth Brazilian Symposium on Neural Networks* (*SBRN’06*), 2006, p. 35.

Michael Hinchey
Associate Editor