

A Neural Network Approach For Long-Term Software MTTF Prediction

Susanta Patra

Philips Research India-Bangalore

susanta.patra@philips.com

Abstract -- Accurate prediction of software MTTF (Mean Time To Failure) is necessary in critical applications. This paper presents the prediction of long-term MTTF using neural network. The proposed method is to learn a sequence of cumulative observation time and the corresponding observed accumulated failures up to the present testing time. The results obtained through proposed method were compared with CASRE (Computer-Aided Software Reliability Estimation) tool. It is observed that information provided by the proposed method is more meaningful as compared to existing methods.

I. INTRODUCTION

For many critical applications, software can be released only after some prescribed reliability criterion has been satisfied at the end of its development and testing cycles. In such cases accurate prediction of reliability is necessary.

Neural network [2] has been applied to estimate parameters of the formal model and to learn the process itself in order to predict the future outcomes. It has been shown that feed-forward network can be applied for prediction.

Back-error propagation is one of the most widely used neural network paradigms and has been applied successfully in application studies in a broad range of areas [1]. We have proposed software reliability prediction method using back-propagation neural network. Objectives of the work are:

1. To propose NN method for long-term prediction of reliability, MTTF, PPM (Parts Per Million) and software failures.
2. To compare the existing methods from CASRE tool and proposed NN method on the basis of Kolmogorov-Smirnov test and different error methods.

II. LONG-TERM RELIABILITY PREDICTION

Proposed method has used back-propagation neural network algorithm [1] for cumulative execution time as an input and cumulative number of failures as output. We considered one input and one output neural network. The method is used for long-term prediction of MTTF, reliability, number of failures and PPM. The reliability indices are defined below.

Using Chi square distribution, MTTF is defined by
$$MTTF = 2 * T / \chi^2(1-C.L., 2 * N + 2),$$

where C.L.=Confidence level,

N= Number of failures

T = Execution time

Reliability after 1 hour of operation = $\text{Exp}(-1/MTTF)$

PPM (the expected number of failures over the first hour of operation of 1,000,000 software/systems)
 $= 1000000 * (1 - \text{Exp}(-1/MTTF))$

The proposed method is compared with meaningful measures [2] in following analytical models using CASRE tool[2].

- NHPP (The nonhomogeneous Poisson process)
- S-Shaped Model
- Poisson and Binomial Models

III. RESULTS AND DISCUSSIONS

We will see how different software reliability growth models can be compared for their predictive accuracies. We have used software test data [2] as shown in Table 1.

For the test data, we have fitted NHPP, S-Shaped, Poisson and Binomial using CASRE tool and proposed method. We have analyzed these results on the basis of –

1. Root Mean Squared Error (RMSE) = $(\sqrt{\sum_n (p-a)^2})/n$, where n=# of observations, p=predicted output, a=actual output
2. Mean Absolute Error (MAE) = $1/n * \sum_n |p-a|$
3. Mean Absolute Percent Error (MAPE) = $1/n * \sum_n (|p-a|/a) * 100$
4. Correlation Coefficient (Cor. Coef.) = $(\sum_n (p - \text{mean}(p))(a - \text{mean}(a))) / \sqrt{(\sum_n (p - \text{mean}(p))^2 \sum_n (a - \text{mean}(a))^2)}$
5. Kolmogorov-Smirnov (K-S) test. The procedure is:
 - Tabulate the ranked failure data. Calculate the values of $|a_i - m_i|$ where a_i is the i th actual cumulative failures and m_i is the i th modeled cumulative failures. All a_i and m_i must be normalized.
 - Determine the highest single value.
 - Compare the value with the appropriate K-S value.

The errors and correlation coefficient calculated for the five different models are shown in Table 2. Smaller the error, higher is the prediction accuracy and higher the correlation coefficient, higher is the prediction accuracy. Thus, the values of errors and correlation coefficient clearly indicate that proposed tool provides better fit than the parametric models. The K-S test is carried out for all five models. The results

TABLE 1
TIME TEST DATA[2]

Cumulative Execution Time(Hrs)	Cumulative Number of Failures	Cumulative Execution Time(Hrs)	Cumulative Number of Failures
56	14	840	174
112	33	896	177
168	56	952	178
224	68	1008	180
280	90	1064	180
336	102	1120	182
392	115	1176	191
448	134	1232	192
504	144	1288	192
560	149	1344	192
616	154	1400	192
672	159	1456	193
728	166	1512	194
784	173		

TABLE 2
SUMMARY OF ERRORS FOR DIFFERENT MODELS

Prediction Errors	Model Used				
	NN	NHPP	S-Shaped	Poisson	Binomial
RMSE	0.483	1.013	1.233	1.288	1.167
MAE	2.00	4.159	5.053	4.522	4.000
MAPE	2.047	6.79	7.13	8.001	7.372
Cor. Coef	0.99884	0.99731	0.99768	0.99844	0.99841
	Actual K-S Test				
	0.0206	0.0604	0.0815	0.0868	0.0806
	10 percent s-significant level: 0.234				

obtained are presented in Table 2. We have used 10 percent statistical significance level. We found that critical value from Kolmogorov-Smirnov test table greater than the actual K-S values. Therefore the null hypothesis, i.e. the data do fit the assumed distribution at 10 percent s-significant level, is rejected. The calculated K-S values indicated that proposed model is the best.

Figure 1 shows the long-term prediction curve using neural network. We used the cumulative execution time 1512 hours and cumulative failures 194. It is clear from the figure that the predicted failures are 199 up to the execution time of 50,000 hours. MTTF is approximately 223 hours in 95% confidence level as shown in Fig.2.

IV. CONCLUSION

From the results of comparison it is clear that the proposed NN method provides better fit than parametric models. The major advantages in using neural network approaches are:

- Easy construction of models of varying complexity at different points of time for a given data set.
- Easy adaptation of models of varying complexity to different test data sets.
- Easy evaluation of model parameters (i.e., finding weights of the network) using network training algorithm.

We have demonstrated the long-term prediction of software failure and MTTF for software test results.

REFERENCES

- [1] K. Mehrotra, C.K. Mohan and S. Ranka, "Elements of Artificial Neural Networks", *Penram International Publishing*, 1997.
- [2] M.R. Lyu, "Handbook of Software Reliability Engineering", *IEEE Computer Society Press*, McGraw Hill, 1996.

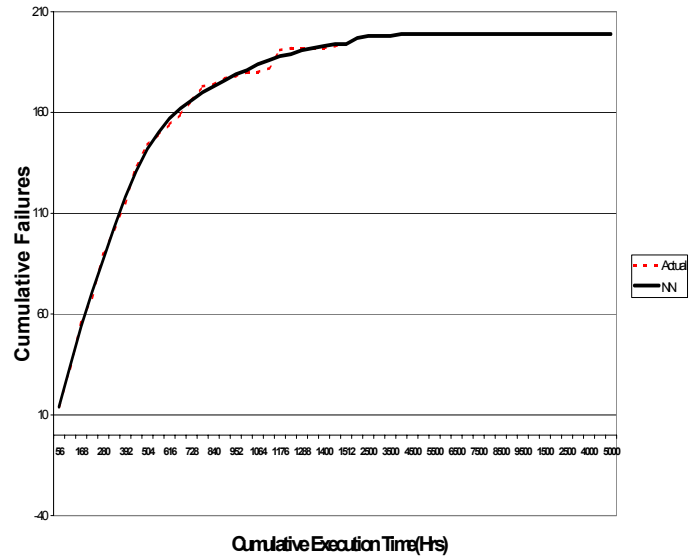


Fig.1: Long-Term Failure Predictions Using NN

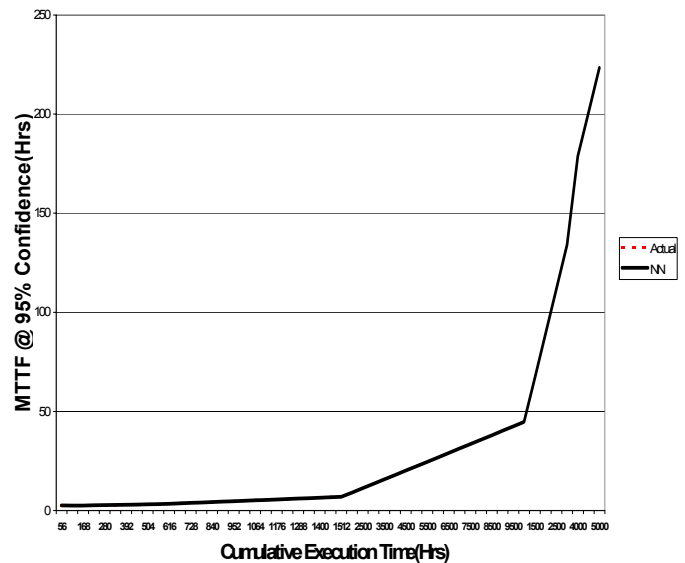


Fig.2: Long-Term MTTF Predictions Using NN