TATA CONSULTANCY SERVICES

TATA

Simulation & Neural Network Applications in Software Reliability Prediction

By Vojo Bubevski

26 April, 2010

Experience certainty. IT Services
Business Solutions
Outsourcing

Agenda (Software Reliability Prediction)

- Introduction
- The Project & Data
- Simulation & Neural Network Models
- Analytic Models
- Verification of Results & Comparison
- Conclusion
- References
- Questions & Answers

Introduction: Terminology

- **Monte Carlo Simulation:** Methodology for evaluating a deterministic model by applying a sequence/distribution of random numbers as inputs;
- **Neural Networks:** Computational methods based on mathematical models of biological nervous systems; Operate on principle of learning; No a priory model is required.
- **Software Reliability:** The probability of failure-free software operation for a time-period;
- **Software Reliability Engineering (SRE):** Studies the software reliability;
- **Software reliability prediction:** Determines future reliability based upon software metrics data collected (available).
- **Cumulative Failure Function:** The mean cumulative failures at each point in time;
- **Failure Intensity Function:** The rate of change of the cumulative failure function;
- **Mean Time to Failure (MTTF):** The expected mean time that the next failure will occur;
- **Defect:** Nonconformity of software to specifications (**fault** – cause, or **failure** – effect);

Introduction: Subject Context

- Software Reliability is a key customer-oriented measure in Six Sigma Software;
- Classical Approach to Software Reliability Prediction: Analytic models since 1970s;
- Analytic models characteristics:
  - Static – do not account for the inherent variability and uncertainty of the software processes;
  - Unrealistic and oversimplified assumptions required to obtain a simple analytic solution;
- Simulation application in Software Reliability Engineering since 1993;
- My work: Simulation & Neural Network models in Software Reliability Prediction;
- Simulation models:
  - Account for the uncertain and dynamic factors involved in the software processes;
  - Simulation models are subject to only a few fundamental assumptions;
  - Provide for more accurate modeling than can be obtained from analytic models;
- Neural Network models:
  - Neural Networks are linked to nonparametric models;
  - No need for estimation of parameters and a priory models.

Galileo Project

- Galileo spacecraft Control and Data System (CDS) project at Jet Propulsion Laboratory™ using published data;
  - Actual data available for entire testing phase (41 weeks) of the project;
  - Only data from the fist 23 weeks of testing are used in the experiments;
  - The rest of the data are used for verification of experimental results.
  - Actual schedule:
    - Operation Testing from Week 1to Week 41;
    - System delivered for operation in Week 42.
  - Assumption to emulate on-going project:
    - Operation Testing planned to be completed at the end of Week 41;
    - Project is on schedule at the end of Week 23;
    - Targeted delivery date in Operation is Week 42;

Galileo Project Data: Control and Data System (CDS)

Actual (Raw) Data

- Actual Failure Intensity Function exhibits a zigzag decreasing behavior.
- Actual Cumulative Failure Function is increasing but not smoothly;
- Data are raw – Failure Intensity Function & Cumulative Failure Function are not practical for simulation;

Method to Transform Raw Data for Simulation

- The numbers of failures detected in each time interval are independent Poisson random variables.
- The raw data time intervals are sorted in descending order of the number of failures in each interval, preserving the Software Reliability principles: a) The number of failures per time interval; b) The time period; c) The total number of defects detected during the time period; and d) MTTF.

Transformed Data

- Transformed Failure Intensity Function (Figure 1);
- Transformed Cumulative Failure Function (Figure 2).

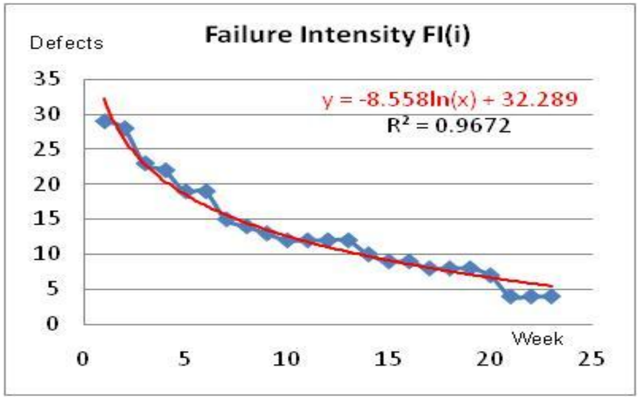

Figure 1: Transformed Failure Intensity Function

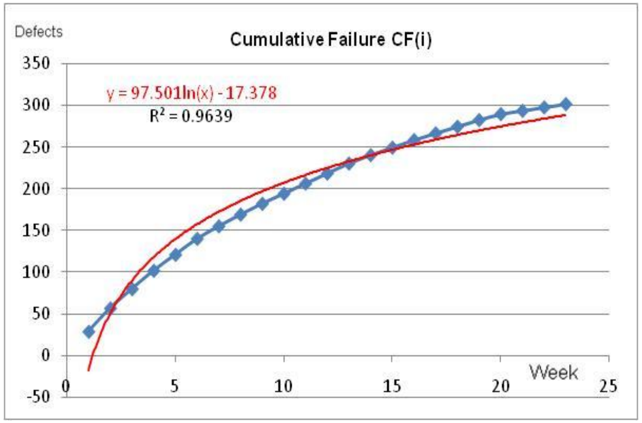

Figure 2: Transformed Cumulative Failure Function

Software Reliability Prediction: The Models

<u>Two software reliability models are used:</u>

- Logarithmic Poisson Reliability Model: This model is used to predict the future course of the *Failure Intensity Function (FIF)* with a Logarithmic Approximation;

- Reliability Growth Model: This model is used to predict the future course of the software reliability growth curve using *Cumulative Failure Function* with a Logarithmic Approximation.

<u>**Note:** For practical reason, only the best Simulation Model & the best Neural Network Model are presented.</u>

Simulation – Logarithmic Poisson Reliability Model (LPRM)

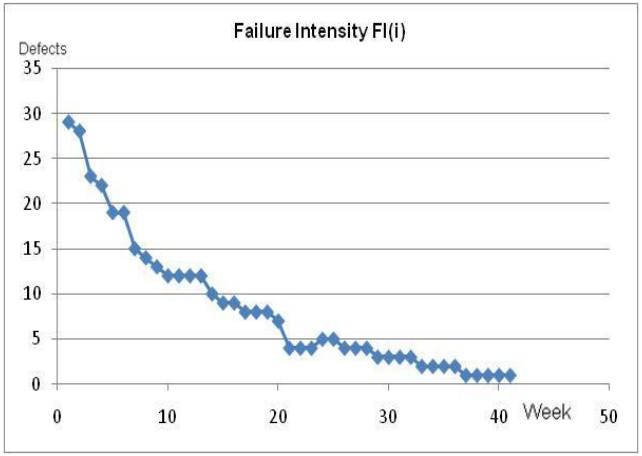

Figure 3: Failure Intensity – Actual & Predicted (LPRM)

Table 1: Predicted Cumulative Defects (LPRM) Week 41

Testing Process Description	Total Defects Mean, μ	Standard Deviation σ	Min Value	Max Value
Galileo CDS	348	6.84	326	385

Simulation – Prediction of Failures in Testing Week 24 - Week 41

- Discrete event simulation based on the logarithmic Poisson software reliability model.
- Poisson Distribution Mean equals the value of the logarithmic Failure Intensity Function for time *t* (Fig. 1).

Simulation results (Fig. 3 & Table 1):

- The actual Failure Intensity for Week 1 – Week 23 (Fig. 3);
- The predicted (simulated) Failure Intensity for Week 24 – Week 41 (Fig. 3).
- Predicted Cumulative Defects (LPRM) Week 41 (Table 1)

Neural Network – Logarithmic Poisson Reliability Model (LPRM)

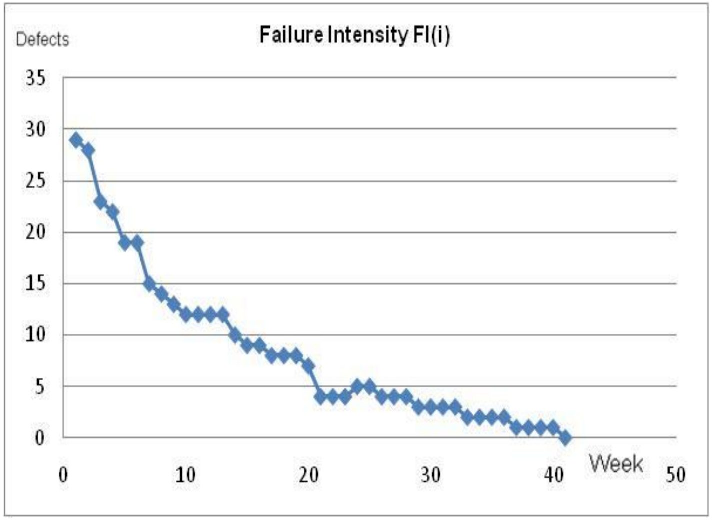

Figure 4: Failure Intensity – Actual & MLFN Predicted (LPRM)

Table 2: MLFN 6N Predicted Cumulative Defects (LPRM) Week 41

Testing Process Description	Total Defects Mean (μ)	Standard Deviation Abs. Error	RMS Error	Mean Absolute Error
Galileo CDS	347	0.06	0.07	0.03

<u>Neural Network – Prediction of Failures in Testing Week 24 – Week 41</u>

- Multi Layer Feed-forward Network (MLFN) based on the logarithmic Poisson software reliability model.
- MLFN 6 Nodes trained on the values of the 23 weeks logarithmic Failure Intensity Function for time *t* (Fig. 1).

<u>Prediction results (Fig. 4 & Table 2):</u>

- The actual Failure Intensity for Week 1 – Week 23 (Fig. 4);
- The predicted Failure Intensity for Week 24 – Week 41 (Fig. 4).
- MLFN 6N Predicted Cumulative Defects (LPRM) Week 41 (Table 2)

Simulation – Reliability Growth Model (RGM)

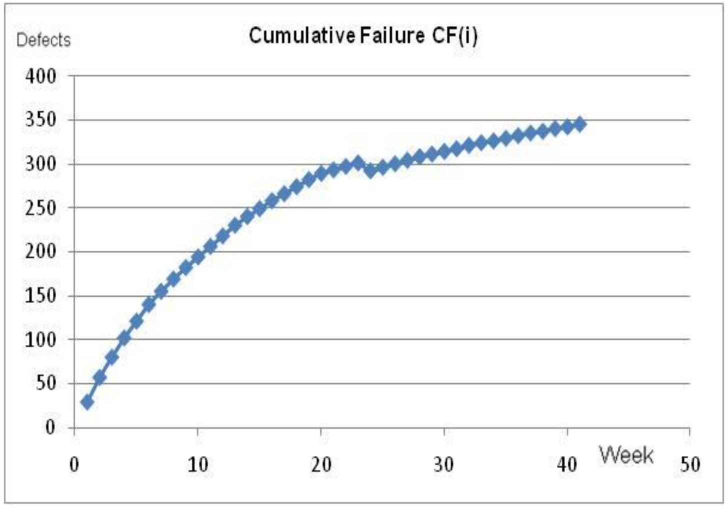

Figure 5: Cumulative Failure – Actual & Predicted (RGM)

Table 3: Predicted Cumulative Defects (RGM) Week 41

Testing Process Description	Total Defects Mean, μ	Standard Deviation σ	Min Value	Max Value
Galileo CDS	345	18.61	274	422

Simulation – Prediction of Failures in Testing Week 24 – Week 41

- Discrete event simulation based on the Reliability Growth Model.
- Poisson Distribution Mean equals the value of the logarithmic Cumulative Failure Function for time *t* (Fig. 2).

Simulation results (Fig. 5 & Table 3):

- The actual Cumulative Failure for Week 1 – Week 23 (Fig. 5);
- The predicted (simulated) Cumulative Failure for Week 24 – Week 41 (Fig. 5).
- Predicted Cumulative Defects (RGM) Week 41 (Table 3)

Neural Network – Reliability Growth Model (RGM)

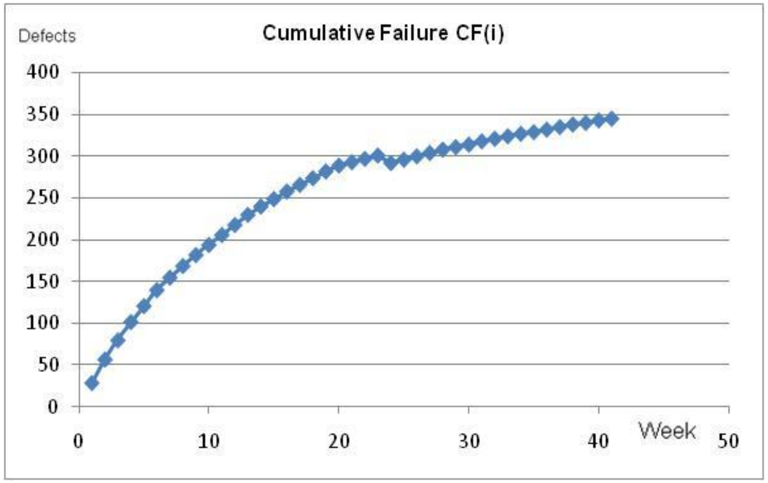

Figure 6: Cumulative Failure – Actual & MLFN Predicted (RGM)

Table 4: MLFN 6N Predicted Cumulative Defects (RGM) Week 41

Testing Process Description	Total Defects Mean (μ)	Standard Deviation Abs. Error	RMS Error	Mean Absolute Error
Galileo CDS	345	0.0011	0.0013	0.0008

<u>Neural Network – Prediction of Failures in Testing Week 24 – Week 41</u>

- Multi Layer Feed-forward Network (MLFN) based on the Reliability Growth Model.
- MLFN 6 Nodes trained on the values of the 23 weeks logarithmic Cumulative Failure Function for time *t* (Fig. 2).

<u>Prediction results (Fig. 4 & Table 2):</u>

- The actual Cumulative Failure for Week 1 – Week 23 (Fig. 6);
- The predicted Cumulative Failure for Week 24 – Week 41 (Fig. 6).
- MLFN 6N Predicted Cumulative Defects (LPRM) Week 41 (Table 4)

Analytic Models & Simulator Results

Galileo CDS data were used for reliability prediction by means of analytic models and a simulation model using a special-purpose simulator. The results of this reliability prediction work have been published [1].

- The simulator results of Galileo CDS Testing are given in [1], Sec. 16.7.2, Table16.3. The total number of defects predicted by the simulator for the Galileo CDS testing is 341 (cumulative defects at the end of testing).
- The results of three analytic models are also given in [1], Sec. 16.7.2, Figure 16.12.

The following are the results of the three analytic models and the associated errors of the readings from the graph.

1. Jelinski – Moranda (JM) Model: Predicted Total 381 Defects, Reading Error is 0.82% (i.e. 3 defects);
2. Musa – Okumoto (MO) Model: Predicted Total 367 Defects, Reading Error is 0.82% (i.e. 3 defects);
3. Littlewood – Verrall (LV) Model: Predicted Total 614 Defects, Reading Error is 0.5% (i.e. 3 defects);

Verification of Results & Comparison

- The published results of the Galileo software system reliability prediction given above are used to evaluate the results of the Simulation and Neural Networks models in this paper.
- Table 5 shows the evaluation and comparison of the models.
- The comparison shows that the Simulation and Neural Networks' models are superior to both the analytic models and the simulation model using the special-purpose simulator.
- Simulation and Neural Network models based on the Logarithmic Poisson Reliability model are better than the respective models based on the Reliability Growth Model. Also, the Simulation LPRM is slightly better than the MLFN LPRM.
- Apparently, the best predicting model for Galileo CDS reliability is the Simulation LPR Model.

Table 5: Software Reliability Prediction Models' Evaluation & Comparison

Prediction Model Description	Total Defects Predicted	Total Defects Actual	Model Absolute Error	Model Percent. Error
JM	381	351	30	8.55%
MO	367	351	16	4.56%
LV	614	351	263	74.93%
Simulator	341	351	-10	-2.85%
MLFN LPRM	347	351	-4	-1.14%
MLFN RGM	345	351	-6	-1.71%
Simulation LPRM	348	351	-3	-0.86%
Simulation RGM	345	351	-6	-1.71%

Conclusion: Summary

- The paper presented Simulation and Neural Networks applications in Software Reliability Prediction.
- The models were applied to a real software project using published data, i.e. Galileo CDS testing project at the Jet Propulsion Laboratory.
- The purpose was to predict the system's reliability at the end of the testing phase.
- Different models were applied in order to evaluate the models;
- Simulation & Neural Network predictions were compared with the published results from analytic models of Galileo CDS reliability prediction.

Conclusion: Major Points to Emphasise

1. Conventional analytic models are parametric thus requiring the estimation of parameters. Also, unrealistic and oversimplified assumptions are required to obtain a simple analytic solution.
2. Simulation models appear to be subject to only a few fundamental assumptions;
3. No assumptions nor a priory models required for Neural Networks.
4. Simulation and Neural Networks are superior to the analytic models.
5. The Simulation and Neural Networks superiority to the conventional analytic models was proved by the comparison of the results of the presented models, and the published results of the analytic models.
6. The Galileo CDS published testing data used in the experiments;
7. The data were reliable as verified at the Jet Propulsion Laboratory.
8. Quality of data is critical success factor as the methodologies are data-driven.
9. Microsoft™ Excel® and Palisade™ @RISK® & NeuralTools® were used;
10. Palisade™ @RISK® & NeuralTools® are comprehensive and easy to use.

References

1. Lyu, Michael R., *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, 1996.
2. **Stephen H. Kan, *Metrics and Models in Software Quality Engineering*, Addison-Wesley Professional, September 26, 2002.**
3. Von Mayrhauser, A., et al., "On the need for simulation for better characterization of software reliability", Proceedings, Fourth International Symposium on Software Reliability Engineering, 1993.
4. Gokhale, Swapna S., Lyu, Michael R., Trivedi, Kishor S., "Reliability Simulation of Fault-Tolerant Software and Systems", Proceedings of Pacific Rim International Symposium on Fault-Tolerant Systems, 1997.
5. Gokhale, Swapna S., Lyu, Michael R., Trivedi, Kishor S., "Reliability Simulation of Component-Based Software Systems", Proceedings of Ninth International Symposium on Software Reliability Engineering, 1998.
6. Tausworthe, Robert C., Lyu, Michael R., "Software Reliability Simulation" Chapter 16, Handbook of Software Reliability Engineering, IEEE Computer Society Press, 1996.
7. Bubevski, Vojo, "An Application of Simulation in Software Reliability Prediction", 2008 Palisade™ Risk & Decision Analysis Conference, New York City, 13 – 14 November, 2008.
8. Karunanithi, N., Malaiya, Y.K., "Neural Networks for Software Reliability Engineering" Chapter 17, Handbook of Software Reliability Engineering, IEEE Computer Society Press, 1996.
9. Karunanithi, N., Malaiya, Y.K., Whitley, D., "Prediction of software reliability using neural networks", Software Reliability Engineering, 1991, Proceedings, International Symposium on, 1991.
10. Aljahdali, S.H. Sheta, A. Rine, D., "Prediction of software reliability: a comparison between regression and neural network non-parametric models", Computer Systems and Applications, ACS/IEEE International Conference on, 2001.
11. Khoshgoftaar, T.M. Allen, E.B. Hudepohl, J.P. Aud, S.J. , "Application of neural networks to software quality modeling of Avery large telecommunications system", Neural Networks, IEEE Transactions on, Volume: 8, Issue: 4, 1997.
12. Sitte, R., "Comparison of software-reliability-growth predictions: neural networks vs. parametric-recalibration", Reliability, IEEE Transactions on, Volume: 48, Issue: 3, 1999.
13. Paul Bratley, Bennett L. Fox, Linus E. Schrage, *A Guide to Simulation*, Springer-Verlag, New York, 1983.
14. Reuven Y. Rubinstein, Dirk P. Kroese, *Simulation and the Monte Carlo Method*, John Wiley & Sons, New Jersey, 2008.

Questions & Answers

Thank You