TechTalks

Enhancing Reliability Modeling of Software using ANN Approach

Prof. (Dr.) Sunil Kumar Khatri
Professor and Director
Amity Institute of Information Technology,
Amity University Uttar Pradesh, Noida, U.P., India
sunilkkhatri@gmail.com, skkhatri@amity.edu

There are numerous instances where failures of computer-controlled systems have led to colossal loss of human lives and money. With increased complexity of products design, shortened development cycles and highly destructive consequences of software failures, a major responsibility lies in the areas of Software Debugging, Testing and Verification. As software systems have become more and more complex, the importance of effective, well planned testing has increased many folds.

Several software reliability growth models designed to work in different environments exist in the literature. As projects were developed, new software reliability growth models were built to suit them. Consequently, we have a large number of SRGM, however none of which works across different environments.

Software reliability is a key part in software quality. The study of software reliability can be categorized into three parts: modeling, measurement and improvement.

As more and more software is creeping into embedded systems, we must make sure they don’t embed disasters. If not considered carefully, software reliability can be the reliability bottleneck of the whole system. Ensuring software reliability is no easy task. As hard as the problem is, promising progresses are still being made toward more reliable software. During software testing, Fault Removal Rate (FRR) does not remain constant. The introduction of change factor in FRR helps in better predictability and more accuracy.
It has been observed that number of factors such as software development methodology, software development environment, complexity of the software, software personnel, etc. may affect the behavior of software reliability. Recently, artificial neural network approach has been applied in estimating and predicting software reliability growth phenomenon. Neural networks models have a significant advantage over analytical models, because they require only failure history as input and no assumptions. In reply to this, neural network approach has been applied to software reliability assessment.

A number of factors that normally demonstrate non-linear patterns such as software development methodology, software development environment, complexity of the software, software personnel, and etc. may affect the behavior of software reliability [1]. This imposes several limitations on existing statistical modeling methods that depend highly on the assumptions. Neural networks models have a significant advantage over analytical models, because they require only failure history as input and no assumptions. Consequently, they have drawn attention of many researchers in recent years. It has been found that neural network methods can be applied to estimate the number of faults and predict the number of software failures as they often offered better results than existing statistical analytical models.

As reliability growth models exhibit different predictive capabilities at different testing phases both within a project and across projects, researchers are finding it nearly impossible to develop a universal model that will provide accurate predictions under all circumstances. A possible solution is to develop models that don’t require making assumptions about either the development environment or external parameters. Recent advances in neural networks show that they can be used in applications that involve predictions. Neural network methods may handle numerous factors and approximate any non-linear continuous function.

Many papers are published in the literature addressing that neural networks offer promising approaches to software reliability estimation and prediction. Karunanithi et al. [2, 3 and 4] first applied neural network architecture to estimates the software reliability. They also illustrated the usefulness of connectionist models for software reliability growth predictions. Cai et al. [5] used the recent 50 inter-failure times as the multiple-delayed-inputs to predict the next failure time and found the effect of the number of input neurons, the number of neurons in the hidden layer and the number of hidden layers by independently varying the network architecture. They advocated the development of fuzzy software reliability growth models in place of probabilistic software reliability models.
Sherer [6] has applied neural networks for predicting software faults in several NASA projects. Khoshgoftaar et al. [7] used the neural network as a tool for predicting the number of faults in a program and concluded that the neural networks produce models with better quality of fit and predictive quality.

Su et al. [8] have proposed a neural network based approach to software reliability assessment combining various existing models into a Dynamic Weighted Combinational Model (DWCM). Kapur et al. [9] have proposed an ANN based Dynamic Integrated Model (DIM), which is an improvement over DWCM given by Su et al. [8]. Kapur et al. [10] have proposed a Generalized Dynamic Integrated Model (GDIM) using ANN approach, which incorporates the concept of n types of faults.

Khatri et al. [11] have proposed an artificial neural-network based SRGM considering two types of imperfect debugging during fault removal phenomenon. They considered that during a removal attempt a fault might be removed imperfectly. Such a situation results in number of failures being more than number of removals and is known as imperfect fault debugging. Otherwise it may happen that a fault is generated while removing some fault and existence of a generated fault is known only after the perfect removal of original fault. Due to error generation the total fault content increases. Khatri et al. [12] propose an ANN based software reliability growth model based on Ito type of stochastic differential equation.

**Conclusions**

It has been observed that number of factors such as software development methodology, software development environment, complexity of the software, software personnel, and etc. may affect the behavior of software reliability. We have a large number of SRGM; however none of them works well across different environments.

Recently, Artificial Neural Networks have been applied in software reliability assessment and software reliability growth prediction. This technique has a significant advantage over analytical models, because they require only failure history as input and no assumptions.

**Keywords:** Software Reliability Growth Models (SRGM), Fault-Removal Rate (FRR), Artificial Neural Network (ANN), Change-Point, Back propagation
References