Software Reliability Assessment by using Neural Networks with Fuzzy Logic based Systems

Bonthu Kotaiah¹, R.A. Khan²
¹ Research Scholar, Babasaheb Bhimrao Ambedkar University, Department of I.T., Lucknow, India
Email: kotaiah_bonthukclce@yahoo.com
² Associate Professor, Babasaheb Bhimrao Ambedkar University, Department of I.T., Lucknow, India
Email: khanraees@yahoo.com

Abstract— Software is often a key component of the high technology systems that are so common in modern society. High reliable software is critical both to software industries and the end users of the developed software, and also for the social community generally, because failures of software can cause major disruption to business and can even threaten emergency service. It is a challenge to be able to enhance the quality of a model early enough to prevent problems from fault later in the life cycle because it is much more cost-effective to correct software faults early in the development process than later when they cause failure. This is why building software reliability growth models have gained considered importance in assessing reliability of software products. In our paper, we proposed an approach for assessing the software reliability by using different machine learning techniques like neural-networks, fuzzy logic etc., to measure the software errors to improve the Software Reliability at different phases of Software Development Life Cycle(SDLC).

Index Terms— Software Reliability, Neural Networks, Fuzzy Systems, Fault, Software Development Life Cycle(SDLC), Software Reliability Growth Models(SRGMS)

I. INTRODUCTION

Software Reliability is defined as the probability of failure-free operation of the software over a specified period of time in a specified environment[1]. One approach proposed by Brocklehurst et al.(IEEE Trans. on Software Engineering. 1990) is to try a set of models and selecting the one that best suits the situation. This is a trial and error procedure/head and tail procedure for assessing the software reliability at different phases of software Development Life Cycle(SDLC). It was claimed that the different models have different predictive capabilities at different phases of software testing life cycle and there is no single model that can be relied on for accurate prediction in all circumstances for all different phases of Software Development Life Cycle(SDLC) (Whitely et al. IEEE Tran. on Software Engineering. 1992).Whitley et al. (IEEE Tran. on Software Engineering. 1992) specified that,” the problem of selecting a model can be addressed in two ways[2]:

1) By generalizing the applicability of software reliability growth models by analyzing their predictability across a broad spectrum of representative data sets

© Elsevier, 2013
II. PREVIOUS RESEARCH (REVIEW OF LITERATURE)

There was a decades of research conducted and their results on many model structures for the predictive software reliability were assessed and evaluated for consideration for the further research in the field of Software Engineering for the effective assessment of the Software Reliability. Many problems and issues were reported over practicing these model structures that made analysis and design of such model a quite challenging process to complete[3,4,5]. For example:

- Missing or incomplete data
- Large number of variables or unused extra variables.
- Strong co-linearity (cohesion) between/among the variables.
- Complex linear/non-linear relationships between model variables.
- Outliers and small sizes of the data sets used for the model evaluation.

III. ARTIFICIAL NEURAL NETWORKS (ANNs)

To overcome the above problems, we will use machine learning techniques like neural networks, fuzzy-logic approaches for the effective assessment of the Software Reliability. In our research, we combined the advantages of theoretical aspects of Artificial Neural Networks and Fuzzy Inference Systems for software reliability estimation and modeling and we show that it is a promising technique in all aspects when compared to all other existing techniques for assessing the Software Reliability[10].

Artificial Neural Networks approaches found to be useful in many applications. For example:

- Prediction purpose in the fields like weather forecasting.
- System Identification and Control mechanism implementation purpose.
- Computer Network Design and its implementation purpose.
- Image Processing and many others.

A. Applications of Artificial Neural Networks (ANNs)

Software Community

Software Community or industry also realized the advantages of Artificial Neural Networks. They used Artificial Neural Networks as a tool to solve problems in:

- Cost Estimation
- Software Reliability Engineering Strategy
- Software size estimation
- Software quality prediction

An interesting study was conducted on an open-issue of the comparative study for developing predictive models of software metrics for assessing the software reliability was provided (A. Gray and S. MacDonell 1997 in the Information and Software Technology Journal, Elsevier). This study suggested the following techniques for developing the software reliability growth models(SRGMs) for assessing the software reliability at different stages of software development life cycle(SDLC)[6,7].

- Least square regression analysis
- Robust regression analysis
- Neural network
- Fuzzy systems
- Rule based systems
- Case-based reasoning

IV. GOAL

The goal of our paper is to improve accuracy of the Software Reliability Estimate at different stages of Software Development Life Cycle(SDLC)[8,9]. The main objectives of our paper are:

- To provide a comparison among parametric models and non-parametric software reliability growth models.
To develop better software reliability growth models using Artificial Neural Networks and Fuzzy logic based systems.

V. PROPOSED METHODOLOGY

Whitley stated that, “The influence of the external parameters and other peculiarities of a software reliability growth model (SRGM) can be eliminated if we have a system that can develop its own model from the past failure history of the software system” which we can develop very easily by using Artificial Neural Networks and Fuzzy logic based systems.

In our research, we propose an Artificial Neural Network based approach with Fuzzy membership functions as an activation functions for Software Reliability estimation and modeling in the different stages of Software Development Life Cycle (SDLC). Also, we are going to perform comparative study about the performance of some well known existing software reliability growth models against our approach using Neuro-Fuzzy systems in predicting the Software Reliability from the three aspects: goodness of fit, prediction ability for short-term prediction and long-term prediction. The approach has been depicted in figure 1.

This is the methodology we are adopted in our current research. This why we suggest building software reliability growth models using prior collected fault data.

A. Guidelines For The Proposed Approach

Here, the complex problem of software reliability modeling can be split to number of sub-problems:

- Selection of the software reliability model structure
- Estimation of the software reliability model parameters like MTTF (Mean-Time-To-Failure) and Mean-Time-To-Repair (MTTR) etc.,
- Evaluation of the model prediction capabilities.

B. Linear Model Structure

By using the provided data sets (Collected from AT & T Bell Telephone Laboratories) and according to the experimental studies and/or results, It was found that the best Linear model structure which can be developed using Least-Square Regression is the Auto-regression model order 4 shown in figure 2, which will take the 4 inputs ($y_{k-1}$, $y_{k-2}$, $y_{k-3}$, $y_{k-4}$) and process them by using membership functions/activation functions and finally generates a single result ($y_{k}$).

Least Square Regression analysis formula for the above model structure for detecting the predicted faults is:

$$y_k = a_1y_{k-1} + a_2y_{k-2} + a_3y_{k-3} + a_4y_{k-4} \quad \ldots \quad (2)$$

Where in equation 2,

- $y_k$ = Predicted and/or Estimated Faults over the observations.
- $y_{k-1}, y_{k-2}, y_{k-3}, y_{k-4}$ = Previous faults observed during different steps of learning process.
- $a_1, a_2, a_3, a_4$ = Weights attached at hidden layers of the selected model.
We have observed and/or implemented many model structures to predict faults. But, according to experimental results and/or observations, it was found that a regression model of order 4 is the best in our case for assessing the software reliability effectively. Hence, we will consider this one as a foundation for making comparison of different models for assessing their effectiveness in assessing the software reliability.

C. Previous Whitley’s Neural Network Model

- A single-input single-output model was proposed as shown in figure 3. Means only input layer and one output layer for the neural network structure implementation.
- The model input is the execution time $t=1,2,\ldots, N$. $N$ is the number of tests conducted during the software testing process of the Software Development Life Cycle (SDLC) process.
- The model output is the number of the observed faults during the different phases of the Software Development Life Cycle (SDLC) process.

![Figure 3: WHITLEY’S neural network model](image)

Whitley’s NNs Model was build with the use of traditional model structure in mind which count on the running time of the program as a metric for predicting software faults at different stages of SDLC in cumulative manner.

VI. Fuzzy Logic

Fuzzy Logic approach has been successfully used to solve varieties of problems in modeling and identification of nonlinear systems developed by Prof. Lotfi A. Zadeh[11]. Fuzzy Inference System is based on the fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning and is a popular computing framework. In our paper, we combined the advantages of theoretical aspects of Artificial Neural Networks and Fuzzy Inference Systems for software reliability estimation and modeling and we show that it is a promising technique in all aspects when compared to all other existing techniques for assessing the Software Reliability[10].

For the problem under study, it was found that the relationship between the predictive model input and output is nonlinear. Hence, because of this nature, the fuzzy logic concept was used to model the process of effective software reliability assessment process in all the phases of the Software Development Life Cycle (SDLC) process.

A. Advantages Of Fuzzy Logic

1. Fuzzy logic is useful when dealing with incomplete sets of data or crisp data.
2. Fuzzy set membership of data gives greater sensitivity for decision making.
3. Fuzzy logic is useful when dealing with vague (unclear) information.
4. Fuzzy Logic have been used in some preliminary work by the Software Engineering Community as a useful tool for software engineering applications.

For example, we can use a Feed Forward-Neural Network with fuzzy weights (at input layers) with fuzzy membership functions as the activation functions to predict software reliability.

B. A Possible Fuzzy Model Structure

In our work, we propose a regression fuzzy model structure to predict accumulated faults.

1. Using the set of input-output data
2. We specify the number of clusters or groups
3. We specify the number of rules or procedures

Then, we try to estimate the fuzzy model parameters for inputing to the model for the assessment of the observed faults.
VII. Evaluation (Fitness) Criteria

The criteria used for evaluating the selected model and which had been very highly used by the people of the Software Industry especially by the Software Engineering community are:

A. Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE)[14], which can be called as the root mean square deviation, RMSD is an evaluation tool and highly used measure for calculating the difference between the observed values and actual values as a result of the experiment based on some observations of the experiment. All the differences at different stages of the experiments(residuals) are combined together to produce a single measure with the help of this evaluation criteria.

Generally, the RMSE of a model prediction with respect to the estimated variable $X_{model}$ is defined as the square root of the mean squared error and it is shown in the equation 3 given below:

$$
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (X_{obs,i} - X_{model,i})^2}
$$

where $X_{obs}$ is observed values/faults and $X_{model}$ is modelled values at running time $i$.

The RMSE values can be used to differentiate the performance of the different models and to validate the models. Also it will compare the performance of the model with other predictive models as a part of the comparative study. In most of the cases, we can use the Root Mean Square Error (RMSE) for validating the suitability of the selected model to assess the software reliability effectively.

B. Normalized Root Mean Square Error (NRMSE)

Normalized Root Mean Square Error(NRMSE)[14], Non-Dimensional forms of the RMSE are useful because often one wants to compare RMSE with different units, shown in the equation 4 . There are two approaches: normalize the RMSE to the range of the observed data, or normalize to the mean of the observed data.

$$
NRMSE = \frac{RMSE}{X_{obs,max} - X_{obs,min}}
$$

VIII. Possible Results

Feed Forward Neural Networks and Fuzzy logic models perform far better than the traditional software reliability growth models. Feed Forward-Neural Networks and Fuzzy logic models were able to provide models with smaller errors than regression analysis for all kinds of software projects[12].The performances of the developed models were tested on NASA software project data presented in [5] and results are compared with the previous software reliability growth models like Delay S-Shaped model, Goel-Okumoto models. The results shows that our approach based Model have the lowest RMSE values as 0.5774. When the accuracy of the system is calculated then it showed 95.2381% of accuracy. This means that the system is showing very low error and high value of accuracy that is desirable from the prediction system. So, the proposed model based system is recommended to estimate the software effort/reliability for projects.

The effectiveness of the Neuro Fuzzy Inference System was the best option for assessing the Software Reliability[9-10] among all the available hybrid fuzzy systems which is based on the Mamdami method of Adaptive NF Inference system will give you the better learning capability or approximation power[11-12]. For this purpose in MATLAB 7.10.0, the Sugeno based Adaptive Neuro-fuzzy Inference System, which is shown in Figure 4 is used for modeling of software reliability status at different stages of Software Development Life Cycle(SDLC)[13]. The inference system, which is already trained, will get the metric values from the earlier stages of SDLC and estimate the software reliability status of the software components or modules.

The Structure of Adaptive Neuro-Fuzzy Inference System is shown in fig. 4. The NF system is trained using a hybrid learning algorithm using both least squares method and back propagation algorithm. In the forward pass the consequent parameters are identified using least squares and in the backward pass the premise parameters are identified using back propagation[12]. The trained NF
system is then tested for the fifteen inputs and it shows 0.1571, 0.2140 as NRMSE(eqn.2), RMSE(eqn.1) values respectively. The plot of the expected and the output of the NF system for the different inputs is shown in Fig.5.

IX. CONCLUSION

Finally, from the observed results, we will show and prove that the proposed model performs far better than the traditional software reliability growth models in assessing the software reliability and Neuro-Fuzzy Systems approach is more suitable for handling datasets with ‘smooth’ trends than for handling datasets with large fluctuations and the training results are much better than the prediction results. Currently we are investigating the use of recent evolutionary computations to solve the software reliability-modeling problem. The results of the fuzzy logic and neural networks models were very promising. The error difference between the actual and estimated response was small. This finding gives a good indication of prediction capabilities of the developed fuzzy model and neural networks for assessing the software reliability.

X. FUTURE SCOPE

In the future one can follow the flowing directions:
1. We can extend this work to the other machine learning techniques like neuro-fuzzy systems approach, support vector machine approach, self-organizing maps approach, decision-region approach etc. for the better estimation of the software reliability at different stages of Software Development Life Cycle (SDLC) process. We can also incorporate recent evolutionary computational mechanisms for the purpose of assessing the software reliability.

2. Rough Set theory and Taguchi Analysis can be used to find the impact of the different attributes towards Effort Prediction.

3. Simulated Annealing Technique can be used to improve the performance of the Neuro-Fuzzy systems.

4. Accuracy of the Fuzzy Rule Generation for the Fuzzy Inference System can further be improved with decision tree techniques.

ACKNOWLEDGEMENT

The authors wish to thank University Grants Commission, New Delhi, Govt. of India for their support through RGNF (Rajiv Gandhi National Fellowship) to carry out our research work effectively.

REFERENCES


Mr. Bonathu Kotaiah obtained his Bachelor’s degree in Computer Applications from ANU in 2001 and M.C.A from ANU in 2008. During the period from September, 2001 to 2011, he worked as an engineer(L-Cube Innovative Solutions), a Corporate Trainer (SyncSoft & Datapro(Vijayawada), COSIS(Hyd.)),a Computer Programmer(AN University). His research interests include software Engineering, Neural networks,Fuzzy Logic. Presently, he is working as a Full-Time Research Scholar in Babasaheb Bhimrao Ambedkar University(A Central University) in the Dept. of I.T.