An Improved Computational Software Reliability Model using ANFIS

Shubhi Bhardwaj
M.Tech (CSE) Scholar
ABES Engineering College, Ghaziabad (India)

Amit Sinha
Associate Professor
ABES Engineering College, Ghaziabad (India)

ABSTRACT
Software reliability is an important attribute of software engineering to ensure the success of software. Software reliability is the probability that there will no failure for a specified time. The reliability of the software depends on various attributes of software such as Size of software, Number of failures and Total time. These data sets of known software follow a specific trend which needs to be studied. The present work collects and analyzes these data sets. The training of these data sets is done through ANFIS. The relative error at definite epochs is noted. The software to be tested is then passed to same network that will give the desired result.

1. INTRODUCTION AND LITERATURE REVIEW
1.1 Introduction
With the rapid growth of software [4], it is very essential to develop a quality software within resource constraints [8][9]. Software reliability is the process which helps to developed the failure free software which can work failure free operations. Software reliability is the probability in which software will perform failure free operation under specified time of period[8]. Software reliability does not guarantee that the software will be 100% reliable but it will give quantitative measurement of the software. The software reliability depends on some historic data of same domain i.e. reliable and unreliable. The training of such data sets explains the trend of reliability and unreliability. The present work uses such training through ANFIS. The software to be tested when passed upto same training datasets gives more accurate result than previous methods.

1.2 Literature Review
Ramakanta Mohanty et al.[1] proposed a Ant Colony Optimization technique(ACOT) for predicting software reliability based on data collection. The dataset was divided into 80:20 ratio for training and test data respectively. The Root Mean Square Error(NRMS) was used for the measurement of dataset and to verify the result.

Dongliang Yuan et al.[2] proposed a model which uses the ANFIS. The network takes size of software as an input and give the prediction of software as an output. The model accuracy was tested and verify by comparing the predicted result with RBF neural network.

Khyati M.Mewada et al.[5] proposed a framework which include software effort using neuro fuzzy based (ANFIS tool box) of MATLAB. The model implementation was done on 18 NASA dataset. The membership function was computed for each dataset. After computed the membership function, RMSE and MMRE criteria were computed for each membership function for comparing the performance of dataset.

2. OBJECTIVES
The present paper has the following objectives:
(i) To identify the important attributes of different known software.
(ii) To train the data sets and to observe a trend.
(iii) To check the reliability status

3. HYPOTHESIS
The reliability of any software follows a trend of reliability or unreliability and highly depends on three attributes such as size of software (S), number of failures (F), and total time (T)

4. PROPOSED WORK AND METHODOLOGY
The present work uses the trend of reliable and unreliable known software. The earlier research work used the mathematical model having attributes of software. In this work, the problem of checking reliability is reviewed with different methodology and found a significant improvement over the previous one.

The important attributes of the software that may affect the reliability at greater extent are: Size of the software, Number of failure and Total time.

The data collection has been done through several reliable and globally accepted government websites as well as the previous research work over which an improvement is to be claimed such as cyber security and information system analysis center.

The steps for the present work are

4.1 To Identify the Important Attributes:-
The three important attributes are collected from known software. A sample of such list is shown in Table - 1.

<table>
<thead>
<tr>
<th>Size</th>
<th>No. of failures</th>
<th>Time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>415</td>
<td>50000</td>
</tr>
<tr>
<td>124911</td>
<td>86</td>
<td>50000000</td>
</tr>
<tr>
<td>18018</td>
<td>360</td>
<td>50000000</td>
</tr>
<tr>
<td>56200</td>
<td>225</td>
<td>300000</td>
</tr>
<tr>
<td>164500</td>
<td>374</td>
<td>50000000</td>
</tr>
<tr>
<td>39899</td>
<td>381</td>
<td>210000</td>
</tr>
<tr>
<td>204229</td>
<td>390</td>
<td>35000000</td>
</tr>
</tbody>
</table>
4.2 To Train the Set of Attribute and to Fix Trend
ANFIS is adaptive neuro fuzzy inference system, it combines the features of neural network and fuzzy logic principle. ANFIS is a hybrid learning algorithm. It is based on Sugeno-Takagi fuzzy inference system [6].

Matlab toolbox ANFIS is used to train the data sets. The values are divided into two parts for training and testing data sets respectively. Further training error is calculated and fixed at constant error.

4.3 To Check the Reliability Status:-
The software to be tested will be passed through same dataset and observe an error. The error within the set limit will give its reliability.

5. IMPLEMENTATION AND RESULT
The implementation of the present work can be shown through sample data sets
\[
S1 = 21700 \quad F1=136 \quad T1=100000
\]

5.1 Training Error
Training error is calculated through anfis TOOL IN MATIAB R2013A as shown figure 1

![Figure 1: Training Error at epoch 10 is 5.165e-06](image1)

5.2 Testing Error
The given sample is passed through the same training error, shown in figure-2 and found the testing error 1.7731e-06 this error explain that the software is reliable.

![Figure 2: The testing error is 1.7731e-06](image2)
6. VALIDATION OF THE RESULT
The proposed work is applied on the same data sets of [5] and found a good improvement on checking of reliability apart from three attributes in present work in place of one attribute in [5]. The work is again validated through data sets of known software and it is up to 80%.

A sample of such software is shown below:

Table 2: Testing error for different software data sets

<table>
<thead>
<tr>
<th>Size</th>
<th>No. of failure</th>
<th>Time (sec)</th>
<th>Testing error</th>
<th>Status of software</th>
<th>Original status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>415</td>
<td>500000</td>
<td>9.3219e-06</td>
<td>unreliable</td>
<td>unreliable</td>
</tr>
<tr>
<td>124911</td>
<td>86</td>
<td>500000000</td>
<td>8.974e-06</td>
<td>reliable</td>
<td>Reliable</td>
</tr>
<tr>
<td>18018</td>
<td>350</td>
<td>500000</td>
<td>3.4982e-07</td>
<td>reliable</td>
<td>unreliable</td>
</tr>
<tr>
<td>56200</td>
<td>225</td>
<td>300000</td>
<td>9.1151e-06</td>
<td>unreliable</td>
<td>unreliable</td>
</tr>
<tr>
<td>164500</td>
<td>374</td>
<td>50000000</td>
<td>5.3999e-06</td>
<td>reliable</td>
<td>reliable</td>
</tr>
<tr>
<td>39899</td>
<td>381</td>
<td>210000</td>
<td>9.3312e-06</td>
<td>unreliable</td>
<td>unreliable</td>
</tr>
<tr>
<td>204229</td>
<td>390</td>
<td>350000000</td>
<td>3.4895e-07</td>
<td>reliable</td>
<td>reliable</td>
</tr>
<tr>
<td>1991</td>
<td>51</td>
<td>550000</td>
<td>5.4325e-06</td>
<td>reliable</td>
<td>Reliable</td>
</tr>
<tr>
<td>64890</td>
<td>196</td>
<td>310000</td>
<td>9.9496e-06</td>
<td>unreliable</td>
<td>unreliable</td>
</tr>
<tr>
<td>22517</td>
<td>182</td>
<td>1200000</td>
<td>3.1743e-06</td>
<td>reliable</td>
<td>reliable</td>
</tr>
<tr>
<td>56866</td>
<td>117</td>
<td>600000</td>
<td>8.7567e-07</td>
<td>reliable</td>
<td>reliable</td>
</tr>
<tr>
<td>2010</td>
<td>400</td>
<td>500000</td>
<td>6.9597e-06</td>
<td>reliable</td>
<td>Unreliable</td>
</tr>
</tbody>
</table>

7. CONCLUSION
Software has various steps through its development. The reliability of any software is an important aspect. It depends on various features of software. Few of the important features are size, number of failure and time of failure. This paper examines these three features and found definite trend is followed by almost all known software. The implementation in MATLAB through ANFIS gives a constant training error. On the basis of this error, the similar error may be calculated for any unknown software to be tested and can be checked for its reliability.

8. REFERENCES