Neuro-Fuzzy Based Approach to Event Driven Software Testing: A New Opportunity

S. Chaudhury, A. Singhal, O. P. Sangwan

Department of Computer Science & Engineering, AMITY School of Engineering & Technology, AMITY University Uttar Pradesh, Sector 125 NOIDA-201 313 (UP) INDIA
sunayanait@gmail.com

***Abstract*—Event Driven Software (EDS) testing is a very challenging task as a large number of events can be invoked by users. So far it is difficult to test all the user inputs invoked, therefore, test case prioritization is essentially required for giving more priority to test cases which reveal higher faults comparatively. We have proposed test case prioritization for EDS: as the Event Type, Interaction of Event, and Coverage of Event. Priority assigned in the proposed model uses these factors in Adaptive Neuro-Fuzzy Inference System (ANFIS) MATLAB Toolbox based on Neuro-Fuzzy logic model. Evaluation and validation will be done using Average Percentage of Fault Detection (APFD). APFD rate for prioritized sequence using the proposed Neuro-Fuzzy logic model exhibited 81% rate, whereas, non-prioritized test sequences showed70% suggesting, thereby, that after prioritization; rate of fault detection has improved considerably. Data shows that proposed Neuro-Fuzzy logic model is apt for Test Case Prioritization of EDS Testing.**

Keywords—Event Driven Software (EDS); Test Case Prioritization; Event Coverage; Neuro-Fuzzy Mode; Average Percentage of Fault Detection (APFD).

I. Introduction

The quality of software is assured by testing the software development, application and its progression. While testing, test cases are usually created and followed by their execution on an Application Under Test (AUT) [1]. A large number of test cases are used for testing Event Driven Software therefore, we require new testing techniques for undertaking EDS. Owing to the nature of event driven it takes a number of inputs invoked by the user and consequent upon state change it generates a new combination of inputs denoted as the new output. Different outputs are generated for different combination of inputs which EDS generates. It is very tedious job to test all the possible combinations as the growth of the events is exponential. Therefore, execution of all test cases is a cumbersome job, but many times fixing all the bugs can be a very prolonged and may result in delay of the project completion status. Hence, prioritization of the test cases that reveal the maximum faults in the testing process is very important. Hence, prioritizing the test cases is one of the challenging part of research for maintaining QA in Software testing [2,3,4,5]. According to the prioritization techniques for the test cases, test cases are prioritized using priority as a parameter; in other words, it can be defined as a test suite say T, now PT being the number of possible permutations of T, and a function from PT to real numbers which is f [6]. We need to find $T' \epsilon\ PT$, such that $(V\ T'')(T'' \epsilon PT)(T'' \neq T')[F(T') \geq f(T'')]$. In the present paper we have proposed a method where the test cases are being prioritized using three important factors such as: Event Type, Interaction of the Event, and Coverage of the Event by employing ANFIS MATLAB Toolbox. Subsequently, the evaluation and validation will be done using Average Percentage of Fault Detection (APFD).

II. Review Of Literature

The Event Driven Software (EDS) approach was proposed by Xiao-Ling [7] by way of testing technology of Event Driven Software and analyzed the differences between Event Driven Software and the traditional software. It is procedure oriented and based on above analysis the mechanism of event driven and its effect on software testing was unveiled. The event coverage criterion was also defined. The event executing rules such as concurrent event, ordinal event, non ordinal event and predecessor event were described [7] which were followed by method of testing according to the rules.

Bryce and Memon [8] put forwarded interaction coverage which helps in test suite prioritization. The t-way interaction coverage & comparison was performed by fault detection rate criteria for prioritization of GUI based program Test suites. Test suits with m[18aximum coverage of event interaction advantage the maximum, whereas, those having lesser coverage of interaction was a disadvantage as there was no benefit after employing this prioritization technique [8].

Rothermal et al. [9] discussed about the prioritization of test cases for regression testing which increases the effectiveness of test cases for implementation in meeting performance objective. One of it can be the fault detection rate which quantifies how fast a fault is observed during the testing process. Therefore, increased rate of fault detection is much faster and allows the developer in fixing the faults quickly. Prioritization involves one application of regression testing which is retesting once modifications are done in the software. An advantage of prioritization is that the information is comprehended about the past executed test cases in order to get the ordering of the test cases. Several techniques for

978-1-4673-6984-8/16/$31.00 © 2016 IEEE

regression testing were used for extracting useful information by prioritizing test cases [9].

Bryce and Memon [10] postulated testing model for both GUI and web applications. In this model test case prioritization was achieved by employing of interaction based criterion, frequency based on the usage and as well as the count [10]. The data demonstrated that the prioritization through a two-way (criteria based on interaction) and PV-L to S (criteria based on Parameter count) gave the perfect results in detecting the fault rates for the GUI applications. However, main disadvantage lies in the fact that multiple criteria for prioritization are combined to give the final output but in many instances it has been found that multi criteria is more useful rather than considering a single criterion.

Yu and Lau [11] postulated the prioritization using the faults found by the test cases which straight away uses the information regarding ability of fault detection. Based on the proposed fault model interaction between the test cases and the faults is used for test case generation [11].

Huang et al. [12] reported both weighted event flow graph and GUI Test case prioritization, were employed for ranking the GUI test cases which are non-weighted. For assigning weights, events classification is on the basis of their importance throughout the application [12]. Similarly, Gerrard [13] reported testing and prioritization of the GUI applications in which the fault detection effects are dependent on the way of interaction of a given event with another event and requires further exploration. The weight value assigned to will surely have an impression on ability of detecting fault for a given test case.

A. Various factors affecting prioritization of test case

Here assignment of weight value is done by taking the following factors [14] into account:-

- Event Type
- Interaction of the Event
- Coverage of Event

As per earlier reports events can be broadly classified into five major types as Unrestricted-Focus Event, Menu-Open Event, Termination Event, System-Interaction Event and Restricted-focus Event. Based on the importance of each type of events weight value has been assigned to various types of events as reported by Gerrard [13]. For better understanding assignment of weight values of events is presented in Table I.

TABLE I. WEIGHT VALUES ASSIGNMENT TO VARIOUS EVENTS [15]

Type of Events	Weight Value Assigned
Unrestricted-Focus Event	1
Menu-Open Event	2
Termination Event	3
System-Interaction Event	4
Restricted-Focus Event	5

Interaction of Event – Interaction of event in an Event Driven Software directs the program to follow an execution path which is quite different and has the potentiality of divulging new faults in the given system. Here in the anticipated technique the test cases which have high value of the event interaction are assigned the Priority [16].

•**Coverage of Event**- In order to obtain best results test suits which encompasses maximum coverage of event is given additional significance than the test suits providing lesser coverage. Therefore, it will include actions or parameter values as well as the window counts, which gets covered by the test case.

B. Neuro-Fuzzy model proposed for priortiation

The Adaptive Neuro-Fuzzy Inference System (ANFIS) method has been usually employed as an education scheme for Sugeno-type fuzzy system. The number and type of fuzzy system membership functions are defined when applying ANFIS. In the present investigation we have putted forward a Neuro-fuzzy model with different inputs as Event Type, Interaction of the Event and Coverage of the Event. Neuro-fuzzy model proposed has been presented in Fig. 1. The model consists of three major inputs which give distinct Priority values using Rule Base; Genfis1 or Genfis2 command, Epochs & error, followed by ANFIS; stop when Tolerance is achieved. The Neuro-fuzzy system incorporates fuzzy systems having human being like reasoning approach by making use of fuzzy sets; such a model having capability of linguistic in nature comprising of a set of IF-THEN fuzzy rules. Neuro-fuzzy methods are apt for applications, where user interaction in model design or interpretation is desired [17]. Neuro-fuzzy integrated approach infuses the advantages of both artificial neural network and fuzzy logic system. Three different inputs, namely, Event Type, Interaction of the Event and Coverage of the Event which gives distinct Priority Values have been employed. ANFIS is the process of loading training data and generating FIS Fuzzy Inference System (FIS) and then training and testing it against training data. After fuzzification, each output variable undergoes defuzzification. Crisp value as output is received from the fuzzy set as an input [18].

C. Neuro-Fuzzy model proposed for prioritization

The Adaptive Neuro-Fuzzy Inference System (ANFIS) method has been usually employed as an education scheme for Sugeno-type fuzzy system. The Adaptive Neuro-Fuzzy Inference System (ANFIS) method has been usually employed as an education scheme for Sugeno-type fuzzy system. The number and type of fuzzy system membership functions are defined when applying ANFIS.

In the present investigation we have putted forward a Neuro-fuzzy model with different inputs as Event Type, Interaction of the Event and Coverage of the Event. Neuro-fuzzy model proposed has been presented in Fig. 1. The model consists of three major inputs which give distinct Priority values using Rule Base; Genfis1 or Genfis2 command, Epochs & error, followed by ANFIS; stop when Tolerance is achieved. The Neuro-fuzzy system incorporates fuzzy systems having human being like reasoning approach by making use of fuzzy sets; such a model having capability of linguistic in nature comprising of a set of IF-THEN fuzzy rules. Neuro-fuzzy methods are apt for applications, where user interaction in model design or interpretation is desired [17]. Neuro-fuzzy integrated approach infuses the advantages of both artificial neural network and fuzzy logic system. Three different inputs, namely, Event Type, Interaction of the Event and Coverage of the Event which gives distinct Priority Values have been employed. ANFIS is the process of loading training data and generating FIS Fuzzy Inference System (FIS) and then training and testing it against training data. After fuzzification, each output variable undergoes defuzzification. Crisp value as output is received from the fuzzy set as an input [18].

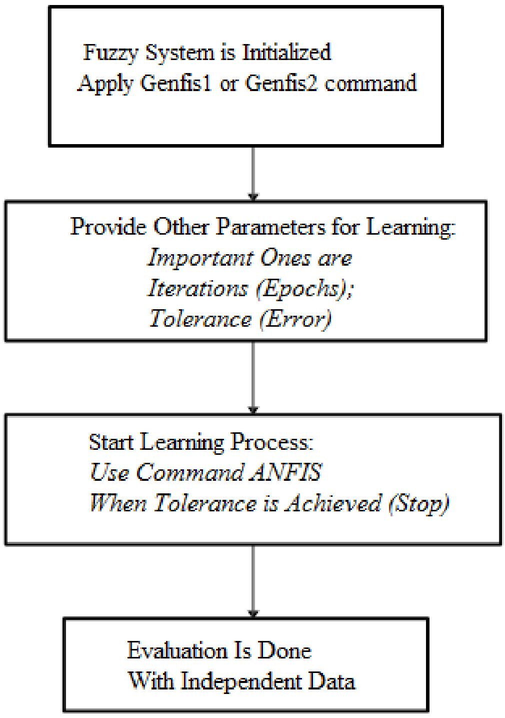

Fig. 1. Proposed Neuro-Fuzzy Model for Prioritization.

III. EXPERIMENTAL DESIGN

Consecutively for fuzzifying the inputs we need a Fuzzy Inference System (FIS) which is generated using Adaptive Neuro Fuzzy System (ANFIS) in which we have two types of data training data and checking data. We prepare a fuzzy system using neural networks using training data. It is a multi-process. In the first step we load the training data and then generate FIS by selecting the number of Member Function (MF's) that are categorized as i.e. very low, low, medium, high and very high and their type and then train FIS by selecting Error Tolerance and Epochs and test FIS against the training data as shown in Fig. 2. The ANFIS model structure is created as shown in Fig. 3 and also the rule viewer and surface viewer can be viewed as shown in Fig. 4.

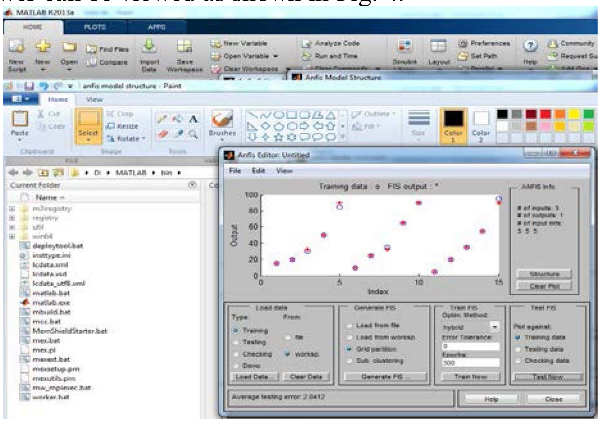

Fig. 2. Testing FIS against Training Data

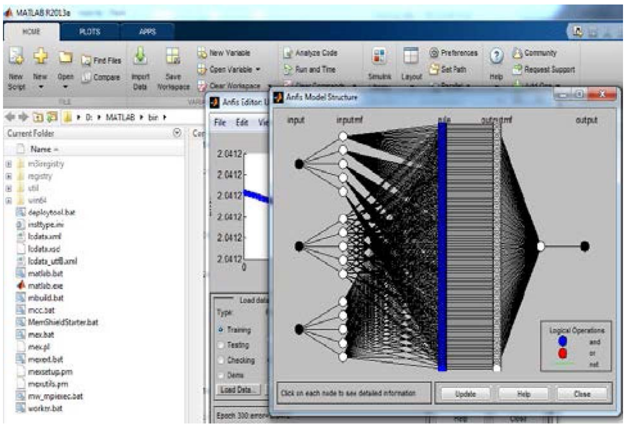

Fig. 3. ANFIS Model Structure

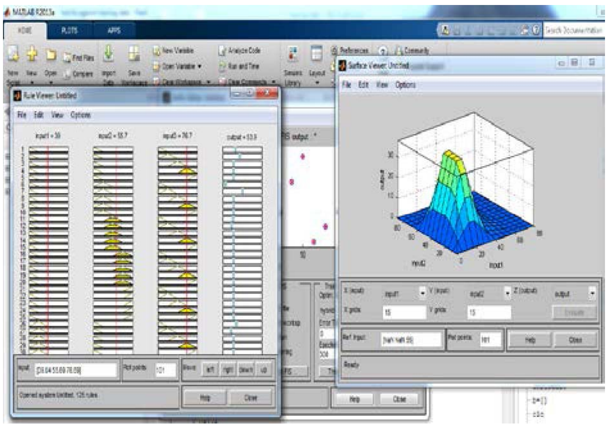

Fig. 4. Rule Viewer and Surface Viewer

As stated above the variables for the input have been categorized. Subsequently, for the output categorization is as low, medium and high.

RULE BASE, EVALUATION PROCESS AND VALIDATION OF MODEL:-

The present model integrates the multiple factors effects like Event Type, Event Interaction of the Event and Event Coverage into a solo parameter which is measurable and will help in defining the priority of test cases, taking into consideration of the chosen knowledge/rule base. All possible input combinations are considered that will create 125 sets. The priority of 125 combinations can be grouped into three domains. Therefore, 125 rules are generated which acts as test cases of the test suite. Among these we will select top ten test cases for further experimentation. For Validation of the model we use checking data which is used to train FIS as shown in Fig. 5 and then test FIS against the checking data shown in Fig. 6.

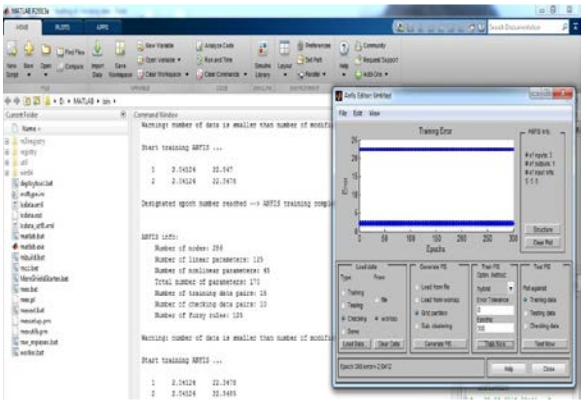

Fig. 5. Checking Data used for Training FIS

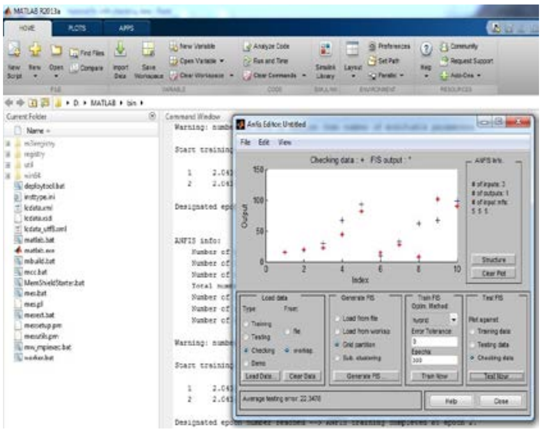

Fig. 6. Testing FIS against Checking Data

IV. EXPERIMENTAL RESULT AND EVALUATION METRIC (APFD)

To the above Neuro-fuzzy model the following crisp value inputs are employed for the fuzzification:- Event Type= 50, Interaction of the Event = 60 and Event Coverage = 90

After fuzzification we saw that the Event Type = 50 belongs to the set Medium. For Interaction of event = 60 belongs to the set High; whereas, Event Coverage = 90 belongs to the fuzzy set Very High. The output that is Priority = 68 we find that according to rules given belongs to fuzzy set High. After getting fuzzified output, the output variable priority has a crisp value [19] by using defuzzification. Hence, we can say defuzzification gives output as a crisp domain from fuzzy domain.

Now let us consider top ten test cases for the prioritization. Suppose our order for prioritized is: {T2,T6,T10,T4,T7,T3,T8,T5,T1,T9} and *non prioritized is : {T1,T2,T3,T4,T5,T6,T7,T8,T9,T10}. After doing prioritization and detection of faults in Event Driven Software, the fault percentage would be evaluated using APFD (Average Percentage of Fault Detection). APFD can be comprehended [20] as:-

Suppose T -> The considered Test Suite

a -> fault count occurring in the software

b -> Test Case count

TFi -> First test position by which faults are exposed.

$$APFD = 1 - \frac{TF1 + TF2 + \ldots + TFm}{ab} + \frac{1}{2a}$$

APFD for Prioritized test case:-

$$= 1 - \frac{1+2+2+3+4+3+3+5+2+1}{10*10} + \frac{1}{20}$$

= .81

APFD for Non Prioritized test case**:-**

$$= 1 - \frac{1+6+4+1+2+4+3+5+4+7}{10*10} + \frac{1}{20}$$

= .70

Thus we can say that the fault detection rate has considerably got better after as 81% is much higher than 70% suggesting, thereby, that Thus the evaluation and validation of our prioritized test cases which was the result of proposed Neuro-Fuzzy model used to prioritize test cases which are large in number created due to successfully achieved events or sequences. This has been possible through user interface by applying prioritization and achieving higher Average percentage of fault detection (APFD).

V. VALIDITY THREATS

Potential validity threats are some of the factors which can impact the results due to prevailing circumstances. First possible threat is the validation of the prioritization of test suite done using the Neuro-Fuzzy model for which we used Average Percentage of Fault Detection for validation of experimental results. Second possible threat may be due to software size because of the stipulations of events and the number of menus that a tester summons through a user interface. Third possible threat to validity may be arising due to variable cost associated with every test case implementation.

VI. Conclusions and Future Work

In present paper prioritization of the test cases was a problem that has been investigated for improving the fault detection rate effectively for Event Driven Software testing. We have putted forward a Neuro-Fuzzy based model for assigning Priority. Priority of test case was categorized into three major domains as either low, medium and high. In the proposed Neuro-Fuzzy Model there are three major factors as Event Type, Interaction of the Event, and Coverage of the Event which are employed for assigning the weight values to the individual test cases. Consequently, impacts of these factors are further categorized into five domains, namely, very high, high, medium, low and very low. Results of the present investigation have been evaluated and validated by employing prioritized test cases which was the result of proposed Neuro-Fuzzy model used to prioritize huge number of test cases which were created due to number of possible events or sequences. This has been possible through user interface by applying prioritization and achieving higher Average Percentage of fault Detection (APFD). Experimental results have revealed that the Neuro-Fuzzy model that is proposed for test case prioritization for Event Driven Software testing has proven to be an effective approach. In future work we may like to explore in designing affordable and cheaper testing model, versatile and having ability to do more prioritization test cases, user friendly, and able to complete the run with precise accuracy, reproducibility and speed for successful execution of multi inputs using test cases by the testers.

Acknowledgment

The first Author wish to acknowledge help and support provided by Department of Computer Science & Engineering, AMITY School of Engineering & Technology, AMITY University, Sector 125 NOIDA, INDIA and both the guides for constant encouragement for completion of research work.

References

[1] G. Paul, "Testing GUI Applications" *EuroSTAR*, Edinburgh UK, 1997.

[2] G. Rothermel, S. Elbaum and A. G. Malishevsky "A group of exact studies: Test case Prioritization," *IEEE Transactions on Software Engineering* vol. 28 (2), pp. 159–182, 2002.

[3] C. Bryce Renee and M. Memon Atif, "Test suite prioritization by communication scope," *Test Automation Workshop On Domain-Specific Approaches to Software* September, 2007.

[4] M. Memon Atif and S. McMaster "Call-Stack scope for GUI test suite decrease," *Software Engineering IEEE Transaction*, vol. 34, Jan/Feb, 2008

[5] M. Memon Atif and S., "Call Stack Coverage of the GUI test-suite diminishment," *Proc., Seventeenth International Symposium on Software Reliability Engineering*, Nov. 2006.

[6] M. Memon Atif and C. Bryce Renee, "Interaction Coverage by Test Suite Prioritization by Test Automation," *Workshop on Domain-Specific Approaches to Software*, Dubrovnik, Croatia, 2007.

[7] D. Xiao-Ling, D. Chun, H. Yong-Hong, "An Approach To Event-Driven Software Testing," Vol. 8.No-4, pp.265-268, 2002.

[8] C. Bryce Renee and M Memon Atif, "Interaction Coverage by Test Suite Prioritization by Test Automation," *Workshop on Domain-Specific Approaches to Software*, Dubrovnik, Croatia, 2007.

[9] G. Rothermel, R. H. Untch, C. Chu and M J. Harrold,"For Regression Testing Prioritizing Test Cases," *IEEE Transactions Software Engineering,* vol. 27, no. 10, pp. 929-948. 2001.

[10] S. Sampath, C. Bryce Renee and M. Memon Atif, "Test prioritization station for occasion driven programming and Building up a solitary model," *IEEE Transaction Software Engineering*, Jan 2010.

[11] Y. Y. Tak and L. M. Fai, "Fault-based test suite prioritization for detail based testing," *Inf. Software Technology* 54, pp. 179-202, February 2012.

[12] C. Y. Huang, J. R. Chang and Y.-H. Chang, "Outline and examination of GUI experiment prioritization utilizing weight-based strategies," *The diary of Systems and Software,* 83, pp 646-659, 2010.

[13] G. Paul, "Testing and prioritization of the GUI applications," Edinburgh UK, 1997.

[14] N. Chaudhary, O. P. Sangwan and Y. Singh, "Experiment Prioritization Using Fuzzy Logic for GUI based Software," *Worldwide Journal of Advanced Computer Science and Applications,* vol. 3, no.12, 2012.

[15] N. Chaudhary, and O.P. Sangwan, "Multi Criteria Based Fuzzy Model for Website Evaluation," In *Proceedings: second International Conference for Sustainable Global Development (INDIACom*). Pp. 1798-1802. 2015.

[16] C. Bryce Renee and M. Memon Atif, "Test suite prioritization by communication scope," *Test Automation Workshop On Domain-Specific Approaches to Software September*, 2007.

[17] A. Nürnberge, "Neuro-Fuzzy Systems,"http://wwwiti.cs.uni-magdeburg.de/~nuernb/nfs/ 2001.

[18] O. P. Sangwan, Y. Singh and P. K. Bhatia "Predicting Software Maintenance Using Fuzzy Model," Published in *ACM SIGSOFT, Software Engineering Notes, USA*, vol. 34, no. 4, pp 1-6, July 2009.

[19] G. Rothermel, H. R. Untch, M. J. Harrold, and C. Chu, "Prioritizing test cases for relapse testing," *IEEE Transactions on Software Engineering,* vol. 27 (10), pp. 102-112, 2001.

[20] T. Kushwaha and O. P. Sangwan, "Prediction of Usability Level of Test Cases for GUI based Application using Fuzzy Logic," In Proceeding *of 4th International Conference CONFLUENCE 2013: The Next Generation Information Technology Summit on the theme: Mega Trends of IT (S.M.A.C), held on 26th-27th Sept. 2013 at AMITY University Uttar Pradesh, Noida, India*" pp. 83-86. 2014.