IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-6 E-ISSN: 2347-2693

A Genetic Algorithm for Regression Test Case Prioritization

Neeraj Kumar Saklani[1]*, Parulpreet Singh[2]

[1]Dept. of Computer Science and Engineering, Baddi University, Baddi, India
[2]Dept. of Computer Science and Engineering, Baddi University, Baddi, India

e-mail: neerajsaklani123@gmail.com, Tel: 19459076329

Available online at: www.ijcseonline.org

Received: 02/May/2017, Revised: 14/May/2017, Accepted: 08/Jun/2017, Published: 30/Jun/2017

***Abstract*—** Regression testing is used to retest the modified version of software. Regression testing is expensive but still an important process. In regression testing, test case prioritization is used to improve the efficiency of the regression test suite by executing the most critical test cases first. As retesting of entire program is not possible with adequate time and cost i.e. only subset of all test cases will execute for regression testing. In this paper, we introduce a technique for regression test case prioritization based on supervised machine learning. We use Genetic Algorithm to make test case description processable for machine learning. In our approach we have consider machine learning classification model logistic regression to evaluate and calculate the prioritization quality. Our result indicates that our technique gives more accurate result as compare to other techniques. We use hybrid combination of genetic algorithm and logistic regression to improve the test case prioritization technique.

***Keywords*—** Regression testing, test cases, prioritization techniques, Genetic Algorithm, Logistic Regression

I. INTRODUCTION

Software testing is a most essential part of software quality assurance. Software testing is a costly and time consuming process but still hold almost 50% of software resources [1]. It is a process to determine error and to find out the difference between the expected result and the actual result. The objective of software testing is to discover minimum number of test cases that can uncover as many defect as possible.

In today's world we are mainly rely on many automatic system and these automatic system must be operate on software. To maintain the quality of software is a difficult task and for this software should be tested before use. So software testing plays an important role in software development life cycle. There are many software testing techniques to perform test on software. Software testing is done to provide reliable and good quality software to the customer.

Regression testing is done to ensure the validity of the changed software [2]. Regression testing is a maintenance activity which is used to determine addition of new functionality does not affect the functionality of the existing software. But due to the addition of new functionality it leads to occurrence of faults and errors that need to be retested. Number of test cases is created during the testing process and it is not efficient to rerun the entire test cases. In regression testing whole test suite is not tested, we select some test cases. Regression testing selects subset of test cases which reduce the size of test suite. In regression test case prioritization testing is scheduled according to some criteria like code coverage and fault detection.

In this paper sections are covered as follows, section I contains the introduction of article, section II contains the background or related work proposed by researchers, section III contains methodology used for implementation, section IV contains the results and discussion, section V contains conclusion and future work.

II. RELATED WORK

There are many regression test case prioritization techniques which are proposed by researchers to solve the test case prioritization problem. Many mechanism have been proposed to solve this problem like fault detection techniques [5], optimal algorithm [7], weight least square method [6]. In 2009, Chen et.al introduced a new hybrid method of edge partition dominator graph & genetic algorithm. In 2011, Andrews et.al introduced a genetic algorithm which use to find out more appropriate test cases. In 2005, Hyunsook et.al introduced the test case prioritization using mutation faults [9]. In 2010, Huang et.al introduced a prioritization technique using genetic algorithm and historic information. Our background survey shows that genetic algorithm is mainly used for optimization and is suitable to implement prioritization techniques.

III. METHODOLOGY

A. Genetic Algorithm

A genetic algorithm (GA) is a form of evolutionary algorithm often used as a search heuristic for combinatorial optimization problems [Bri02]. Invented by John Holland in the 1960s at the University of Michigan, the original goal of genetic algorithms was to formally study the phenomenon of evolution and adaptation as it occurs in nature [Joh92]. A population is formed from a set of "chromosomes" that typically take the form of a string. Usually, the string is a bit string. Each chromosome consists of "genes," such as bits, and each gene is an instance of a particular "allele" (e.g., 0 or 1), where an allele is a biological term denoting the different possible settings of a trait [14]. The GA processes the population of chromosomes by successively replacing one population with another. With each generation, a fitness value is assigned to each chromosome. If the selected chromosomes are strong then they are retained as it is, otherwise different combination are made using crossover operation based on their probability or chromosome are mutated by exchanging individual element of the gene based on their probability. Reproduction continues until the number of chromosomes in the original population is reached for the new population [14]. Then the old population is replaced with the new, and the process is repeated. The fitness-function mainly improves with each generation and this evolution continues until we get an optimal solutions.

The population of individual genes is generated randomly, if knowledge of the solution is known in advance, it can be merged into the first population. Within the determined environment, each individual is tested empirically and assigned a fitness function, which is generally a single number with a higher number representing higher fitness [11].

To find an appropriate fitness function, is one of the most difficult tasks in designing a genetic algorithm. It determines how each chromosome will be interpreted, and it should quantify the optimality of a solution in a genetic algorithm in order to allow that particular chromosome to be ranked against all the other chromosomes. Figure1. Show the execution flow of genetic algorithm.

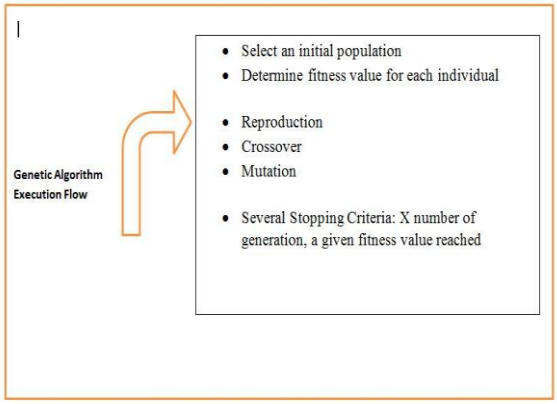

Figure 1. GA execution flow

B. Logistic Regression

The concept of logistic regression was firstly used by Cornfield et.al in 1960's. Thereafter Walter & Duncan2 used this methodology to determine the occurrence function of other variables. Logistic regression is a statistical method which is used to determine the relationship between dependent and independent variable. Dependent variables are those variables whose value we want to predict whereas independent variables are used to influence the dependent variable. The output of logistic regression is calculated with dichotomous variable i.e. there are only two possible outcomes (True & False). The objective of logistic regression is to discover the more suited model which describes the relationship between dichotomous variable i.e. dependent variable describes outcome variable whereas independent variable act like as predictor variable. In term of mathematical equation, logistic regression can predict the logit transformation of the probability of independent variable.

$$\text{Logit}(p) = b_0+b_1X_1+b_2X_2+b_3X_3+\ldots\ldots\ldots+b_nX_n$$

Where p is an probability of independent variable.

The logit transformation is defined as logged odds:

$$\text{Odds}= \frac{p}{1-p}$$

C. Execution Flow of Methodology

In our proposed method we are using machine learning concept to rank test cases. We have use a label dataset which is marked as important and unimportant by the expert as training data. Our main objective is to match tester decision and knowledge. Training data mainly consist of test cases description which is written in natural language. To make these test cases description processable for machine learning we have use genetic algorithm. Genetic algorithm will find out the relevant statement and drop out the non-relevant statement. Suppose there are 100 attribute participating in training data, now many attribute is important is also a matter of concern.

Genetic algorithm will find the gene sequence of 100 attributes and just like real environment it only process the most dominant and healthy gene to successive generation(i.e. most important attribute will take part in machine learning).

Now, to rank test cases based on a given set of training data we have use a regression classification model i.e. logistic regression. Once our classification model has been learned it is applicable for prioritization of test cases. In current practice, it not feasible to prioritize hundreds of test cases

manually. Our approach improves this by automatically prioritize large number of test cases in reasonable time. Test cases are transformed into vector representation and these vector values are fed into the classifier to return a probability value. Thereafter, we order the test cases according to the probability value in descending order and combine them to a new test set.

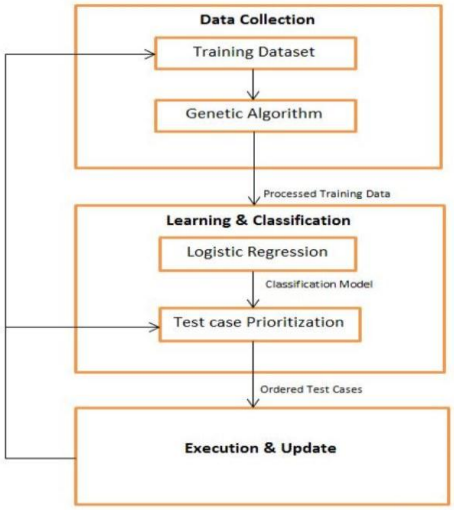

Figure 2. Flow of Proposed methodology

IV. RESULTS AND DISCUSSION

In order to verify the efficiency of regression test case prioritization we use hybrid combination of genetic algorithm and logistic regression which is implemented in Python language. We executed the evaluation program on an Intel core i5 of a 2.30 GHz clock with a Linux operating system.

To determine the performance of regression test case prioritization we have calculated different parameter to justify our findings. Our result shows that false positive value of proposed method is less than existing method which led to increase in accuracy of proposed method. In same way true positive value of proposed value is more as compare to existing method. The results of different parameter are shown in TABLE I.

Parameters	Exiting(SVM)	Proposed(LR)
Mutation Time	100	100
Accuracy	60	89
True Positive	0.55	0.85
True negative	0.45	0.91
False Positive	0.33	0.15
False negative	0.66	0.083
Precession	0.68	0.934
Recall	0.53	0.91

TABLE I. Result of Parameter

The experimental data show that, the accuracy of the proposed method is 89% as compare to the existing method which is 60%. Our proposed method provides more efficient and accurate result which increases the overall performance of regression prioritization technique. The Figure 3.shows comparison of accuracy with existing and proposed method.

Figure 3. Comparison of existing & proposed method

V. CONCLUSION AND FUTURE WORK

We presented a technique for test case prioritization in regression testing. We use genetic algorithm to process test case description written in natural language to make accessible for machine learning. We employ the supervised machine learning technique LOGISTIC REGRESSION to learn a classification model. Our results show considerable improvement as compare to SVM. Our technique is able to find more accurate result and make regression testing more efficient.

For future work, we can use other machine learning classification model for test case prioritization. More investigation needs to be done on selection of training data. We can also work on neural networks or Info- fuzzy network for black-box testing.

REFERENCES

[1] K. K. Ranga, "*Analysis and Design of Test Case Prioritization Technique for Regression Testing*", International Journal for Innovative Research in Science and Technology, vol. 2, pp. 248-252, 2015.

[2] S. Yoo and M. Harman, "*Regression testing minimization, selection and prioritization: a survey*", Software Testing, Verification and Reliability, vol. 22, pp. 67-120, 2012.

[3] T. Hall, S. Beecham, D. Bowes, D. Gray and S. Counsell, "*A systematic literature review on fault prediction performance in software engineering*", Software Engineering, IEEE Transactions on, vol. 38, pp. 1276-1304, 2012.

[4] L.Briand, *"Novel applications of machine learning in software testing",* International Conference of Software Quality, pages 3-10, 2008.

[5] D. Marijan, A. Gotlieb and S. Sen, "*Test case prioritization for continuous regression testing: An industrial case study*", in

Software Maintenance (ICSM), 2013 29th IEEE International Conference on, pp. 540-543, 2013.
[6] G. Chen and P.-Q. Wang. Test case prioritization in a specification-based testing environment. 9(8), 2014.
[7] M. Harman, P. McMinn, M. Shahbaz and S. Yoo, "*A comprehensive survey of trends in oracles for software testing*", University of Sheffield, Department of Computer Science, Tech. Rep. CS-13-01, 2013.
[8] K. Zhai, B. Jiang and W. Chan, "*Prioritizing test cases for regression testing of location-based services: metrics, techniques and case study*", Services Computing, IEEE Transactions on, vol. 7, pp. 54-67, 2014.
[9] J. Petke, S. Yoo, M. B. Cohen and M. Harman, "*Efficiency and early fault detection with lower and higher strength combinatorial interaction testing*", in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, pp. 26-36, 2013.
[10] A. Orso and G. Rothermel, "*Software testing: a research travelogue (2000–2014),*" in Proceedings of the on Future of Software Engineering, pp. 117-132, 2014.
[11] M. J. Harrold, "*Testing: a roadmap*", in Proceedings of the Conference on the Future of Software Engineering, pp. 61-72, 2000..
[12] M. Grindal, B. Lindström, J. Offutt and S. F. Andler, "*An evaluation of combination strategies for test case selection*", Empirical Software Engineering, vol. 11, pp. 583-611, 2006.
[13] G. Rothermel, R. H. Untch, C. Chu and M. J. Harrold, "*Prioritizing test cases for regression testing*", Software Engineering, IEEE Transactions on, vol. 27, pp. 929-948, 2001.
[14] S. Elbaum, A. G. Malishevsky and G. Rothermel, "*Test case prioritization: A family of empirical studies*", Software Engineering, IEEE Transactions on, vol. 28, pp. 159-182, 2002.
[15] M. Gligoric, A. Groce, C. Zhang, R. Sharma, M. A. ALIPOUR and D. Marinov, "*Guidelines for coverage-based comparisons of non-adequate test suites*", Space, vol. 6, pp. 1,142, 2014.
[16] H. Do, S. Mirarab, L. Tahvildari and G. Rothermel, "*An empirical study of the effect of time constraints on the cost-benefits of regression testing*", in Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, pp. 71-82, 2008.
[17] E. Engstrom, P.Runeson, and A.Ljung, *"Improving regression testing transparency and efficiency with history based prioritization- an industrial case study",* in Proceedings of International Conference of Software Testing, Verification and validation, IEEE, pp.367-376, 2011.
[18 A. Perini, A. Susi, and P.Avesani, *"A machine learning approach to software requirement prioritization",*IEEE Transaction of Software Engg, pp.445-461, 2013.
[19] Y.-C. Huang, K.-L. Peng and C.-Y. Huang, "*A history-based cost-cognizant test case prioritization technique in regression testing*", Journal of Systems and Software, vol. 85, pp. 626-637, 2012.
[20] I. H. Witten, E. Frank, and M. A. Hall, *"Data Mining: Practical Machine Learning Tools and Techniques",* Morgan Kaufmann Publishers Inc.,2011.
[21] X. Devroey, G. Perrouin, M. Cordy, P.-Y. Schobbens, A. Legay and P. Heymans, "*Towards statistical prioritization for software product lines testing*", in Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems, p. 10, 2014.
[22] D. D. Nardo, N. Alshahwan, L. Briand and Y. Labiche, "*Coverage-based regression test case selection, minimization and prioritization: a case study on an industrial system*", Software Testing, Verification and Reliability, vol. 25, pp. 371-396, 2015.

Authors Profile

Mr Neeraj Kumar Saklani pursuing master in Software Engineering from Baddi University, Himachal Pradesh. He completed his Bachelor's from Himachal Pradesh Technical University. He has published a paper in IJCSMC journal. His research interest include software testing.

Mr Parulpreet Singh currently working as a Assistant Professor at Baddi University. He has teaching experience of more than 8 years. His research interests include digital image processing.