Transactions of the Nigerian Association of Mathematical Physics
Volume 17, (October – December, 2021 Issue), pp95 – 98

LEVEL OF SIGNIFICANCE DIFFERENCE OF REGRESSION TEST CASE PRIORITIZATION APPROACHES

[1]Samaila Musa, [2]Esther Samuel Alu and [3]Yunana Kefas

[1]Department of Computer Science, Federal University Gusau, Zamfara, Nigeria.
[2,3]Department of Computer Science, Nasarawa State University, Keffi

Abstract

There has been very little evaluations of the GA-based regression test prioritization, even though there are several evaluation methods conducted by so many studies on GA-based regression test prioritization of object-oriented program (OOP), but there little or no approaches that use the level of the significance difference by performing statistical tests to show the significant of the differences between the approaches. The paper conduct a comprehensive empirical study of ten object-oriented programs by the use of Average Percentage of rate of Fault Detection (APFD) to compare HoceDanMafara and one existing software regression tests prioritization together with non-prioritize and random strategies for regression testing of OOP in term of fault detection. The evidence of the statistical test of APFD values of the proposed strategy is shown in the results of the experiment. The study indicated that HoceDanMafara produces significance differences compare with randPrior t, nonPrior and pSherry.

Keywords*; Test case Prioritization; Genetic Algorithm; APFD; Test Case.*

Introduction

Rothermel et al. [1] defined regression test case prioritization strategies is described as a process that allow the software testers to arrange their tests into certain model so that those with the most elevated need are executed before the lower need test case, and it can be utilized in conjunction with tests selection when tests disposing is satisfactory, also it might increase the utilization of testing time more beneficial than non-prioritize when the process of re-executing the test cases is terminated without prior notice.

Fault severity is weight assign to fault by regression test case prioritization used by researchers so that they can be able to use criteria to calculate the total weight of a test case during prioritization[1]. Some work in the literature assigned the same initial weight as the fault severity [2, 3, 4, 5], and some literature used different initial value of fault severity [6]. The total severity of each test case is use to order the test cases, so that those with higher severities are ordered first.

A system level test case prioritization was proposed in order to reveal severer faults at an earlier stage [6]. This is based on factor oriented regression testing using GA. But the changes might not have affected all the test cases, there is a need to select affected test cases and then prioritize them. They also used the same severity of fitness value for a fault even if the fault was executed by the preceding test case. It is realistic to assume that whenever a test case executes a statement in a code, there is decrease in the possibility of an error in that statement [7], therefore, the initial fault severity of the error can be reduced. Most of the previous approaches used APFD values to evaluate their work, by using mean and bar chat. There is need to test for the level of significance differences between approaches before making selection of best among them.

Materials and Methods

There is a need for intensive assessment and investigation/analysis of the adequacy and usefulness of a technique before been considering it as a choice for a new test case prioritization technique. The proposed technique, *HoceDanMafara* is evaluated by conducting a comprehensive experiment aims to quantify the effectiveness of the fault detection of the APFD of the *HoceDanMafara* in comparison with the existing regression test case prioritization approaches as stated in the hypothesis.

Based on the fault detection effectiveness of APFD of the regression testing approaches, the following hypothesis is formulated:

H_0 There is no significant difference in APFD of the prioritize tests effectiveness of *HoceDanMafara* and the three regression testing techniques. H_o: $\mu_1 = \mu_2 = \mu_3 = \mu_4$, where, μ_i is the mean APFD of regression testing approach measured in all the ten objects.

H_A There is a significant difference in APFD of the prioritize tests effectiveness of *HoceDanMafara* and the three regression testing techniques. H_1: $\mu_i \neq \mu_j$ for at least one pair (i, j). One mean is different from the others.

Corresponding Author**:** Samaila M., Email: samailaagp@gmail.com, Tel: +2348065670727

We used R studio to conduct the analysis on the data collected. The validation of the experimental data for the normality assessment is conducted before the real statistical tests. To determine the normality of the data, R studio was use to conduct the validation. The distribution of the experimental data was checked using Shapiro-Wilk test.
The Shapiro-Wilk test compares the scores in the sample to a normally distributed set of scores with the same mean and standard deviation; the null hypothesis is that, sample distribution is normal. If the test is significant, the distribution is non-normal [8].
The test is based on the correlation between the data and the corresponding normal scores [9] and provides the ability to detect whether a sample comes from a non-normal distribution. Some researchers recommend the Shapiro-Wilk test as the best choice for testing the normality of data [10]. It is clear that for effectiveness of the APFD data, the tests have a p-value greater than 0.05, the null hypothesis is accepted, which indicates normal distribution of the data, as shown in Table 1. From the table, the data are normally distributed. A parametric ANOVA test can be used to test the hypothesis.

Table 1. The Shapiro-Wilk Normality Test

	P-value	**Significance**
APFD	0.08872	0.05

Based on the characterized hypothesis and the way that the distribution of experimental data using Shapiro-Wilk test, we used parametric ANOVA test with 5% significance level (i.e. 95% confidence level) in testing the hypothesis. Additionally, in the statistical tests we presumed that the variances of the independent treatments are equal.

Results and Discussion

The data collected from the experiments are presented in our previous research work [5], and shown in Table 2. The APFD of all the regression testing strategies (*HoceDanMafara, randPrior, nonPrior* and *pSherry*) of object-oriented programs is compared and discussed in the table.

Table 2 shows the programs (sample programs), #test cases executed (the first test cases executed by each approach), #total mutants (the total mutants executed), and finally, the APFD. However, in Table 2 we used #Selected test case (number of selected tests) to evaluate the performance of the approaches based on the APFD.

In order to draw conclusions based on the experimental data presented in Table 2, the results are described in respect of APFD. We compared fault detection rate in term of APFD between the regression testing approaches, and also performed hypothesis testing in order to show statistical significance between the regression testing approaches. We compared fault detection rate in term of APFD between the regression testing approaches, and also performed hypothesis testing in order to show statistical significance between the regression testing approaches.

Table 2. Rate of Fault Detection Performance (APFD)

Programs	***NonPrior***	***randPrior***	***pSherry***	***HoceDanMafara***
TrA	58.33	50.44	69.73	78.51
VM1	48.34	69.82	74.26	90.93
VM2	51.84	72.93	79.63	91.84
SLL	70.65	80.77	76.72	89.28
TrS	65.07	69.77	79.01	87.89
BST	68.06	72.87	76.39	83.98
CC1	62.50	75.00	79.16	87.50
CC2	68.12	85.15	89.58	92.71
ATM1	45.01	61.25	70.09	75.00
ATM2	84.19	89.89	87.31	94.48
Average	62.21	72.79	78.19	87.21

The performance of regression testing approach based on average percentage of the rate of fault detection was used to report it performance. In this section, the APFD was studied and reported. To perform this, we conducted an analysis of the APFD of the regression testing approaches of object-oriented programs. We compare the regression test case prioritization approaches (*HoceDanMafara, randPrior, nonPrior* and *pSherry*) by their obtained APFD given in Table 2 as presented using a bar chart as shown in our previous research [5].
The results show that *HoceDanMafara* is more effective than the other three approaches, whereas *pSherry* might be seen as the second effective approach compared to *nonPrior* and *randPrior*. The *nonPrior* approach performed least compared to the three approaches. In order to know the level of the significance difference, we performed statistical tests to show the significant of the differences between the approaches.
The results of APFD of the four regression testing techniques on the ten OOP were presented in Table 2. The results of ANOVA test of data is presented in Table 3.

Based on the decision rule, reject null hypothesis (H_{0apfd}) if F-value > F_{crit} or, equivalently, if p-value < α. From Table 3, we obtained an F-value of 35.306 which was higher than the critical value of 2.96 ($F_{crit} = F_{0.05(3,27)}$) for the F-distribution at 3 and 27 degrees of freedom and 95% confidence for the difference among treatments, as shown in Table 3, so also p-value (1.73e-09) < α (0.05). With respect to the above data, hence, the null hypothesis shall be rejected, H_{0apfd}. With this, we can establish that there are statistically significant differences between a certain number of the regression testing approaches.

Table 3. Results of ANOVA Test on APFD

Source of variation	Degree of freedom	Sum of Square	Mean square	F value	P-value
Between treatment (Prioritization techniques)	3	3276	1092.2	35.306	1.73e-09
Between blocks (Source programs)	9	2306	256.3	8.284	8.39e-06
Residuals	27	835	30.9		
Total	39	6417			

Since there is a significant difference between the approaches, we proceed to test the main effect multiple comparisons using a Tukey's test to the independent variable, and the results are shown in Table 4. The table shows that each approach is compared with the three other approaches. The output indicates that the differences in all the comparisons are significant, except between *randPrior* and *pSherry*. From the table, *HoceDanMafara* compared to the approaches *nonPrior*, *randPrior* and pSherry revealed statistically highly significant, highly significant and significant differences respectively. The approach *pSherry* compared to the *randPrior* and *nonPrior* revealed non-significant and statistically highly significant differences respectively. The approach *randPrior* revealed a statistically significant difference compared with *nonPrior*. This means that there is an evidence to support the alternative hypothesis.

The results in Table 4 (Tukey's test) indicated that most of the evaluations gave outcomes of significant difference between the approaches. In particular, *HoceDanMafara* compared to *pSherry*, *randPrior* and *nonPrior* shown a significant difference. In all the results, *HoceDanMafara* was better (with fewer values) as shown in Table 4. The detailed analysis suggests that *HoceDanMafara* is significantly better than *pSherry*, *randPrior* and *nonPrior* ($p < 0.05$) in terms of APFD. Therefore, from the results, there is enough evidence to conclude that there are statistically significant differences in the APFD of *HoceDanMafara* with *pSherry*, *randPrior* and *nonPrior* regression testing approaches of object-oriented programs.

Table 4. Multiple Comparisons of Significant of APFD

	Diff	Lwr	Upr	P adj
nonPrior- HoceDanMafara	-25.001	-31.807788	-18.194212	0.0000000
pSherry- HoceDanMafara	-9.033	-15.839788	-2.226212	0.0060208
randPrior- HoceDanMafara	-14.423	-21.229788	-7.616212	0.0000205
pSherry-nonPrior	15.968	9.161212	22.774788	0.0000041
randPrior-nonPrior	10.578	3.771212	17.384788	0.0012226
randPrior-pSherry	-5.390	-12.196788	1.416788	0.1582321

The empirical study presented above gives details of the effectiveness of our proposed technique for object-oriented programs in which *HoceDanMafara* was significantly better ($p < 0.05$) than all of the other approaches. Among possible reasons for this outperformance is the ability of regression testing approaches, i.e. the prioritization technique based on the mutants. We kept the object programs and their mutants seeded constant during the experiment.

For the overall, the values from the ten programs for *HoceDanMafara* have the highest APFD scores, followed by *pSherry*. This might be due to using the genetic algorithm in the two approaches. *HoceDanMafara* is better than *pSherry*; this might be due to the reduction of initial fault severity of a fault if executed by the previous test case to a small value compared to *pSherry* where the initial value is not reduced even when a fault was visited by the preceding test cases. *HoceDanMafara* was found to have the highest scores in all the sample programs, as shown in Table 2. The approaches *HoceDanMafara* and *pSherry* on average performed better in versions two of VM, CC and AT (i.e. VM2, CC2, and ATM2), showing that the prioritization approach performed better on larger size test cases and faults than the smaller size. When the number of mutants and test cases are many, applying a prioritization approach will increase the APFD than non-prioritize approach.

The results of the above experiment on APFD show that there was statistically significant difference between prioritization approaches than non-prioritize, and prioritization with reduced severity and same severity when regression testing. *HoceDanMafara* was found to be more effective than the *pSherry* and *randPrior*; this might be due the use of the genetic algorithm in ordering the test cases by reduction of fault severity of the already executed statements by the preceding test cases. The *nonPrior* was found to be less efficient in execution effort; this might be due to non-ordering of the test cases which may result in placing first the less fit test case. This means that if the rate of fault detection is to be considered in regression test case prioritization of test cases, *HoceDanMafara* would be better used for regression testing.

We summarize the discussion that, the proposed technique successfully experimented on realistic object-oriented programs, and shows to be commendable of use as an effective and efficient object-oriented programs test case prioritization technique. We believe that *HoceDanMafara* can be utilized for a complete regression testing process.

Conclusion and Future Work

A regression test case prioritization strategy that ordered selected test cases T` using GA with a reduction in fault severity when a statement is executed by the preceding test cases was proposed, named *HoceDanMafara*. This strategy can be used for regression testing of OOP. A tool was implemented for the *HoceDanMafara*. We prioritized the sample Java programs with the tool. The implementation showed the evidence that *HoceDanMafara* was feasible to be used in practice.

The development of new strategies and the improvement of existing strategies are not the only requirements for regression testing, but there is also a need for evaluations of those strategies and also comparisons between them. *HoceDanMafara* was empirically assessed and compared with other regression testing strategies by the use mutation analysis and APFD metric. The goal of the experiment is to statistically compare *HoceDanMafara* with non-prioritize after selection (*nonPrior*), random prioritization (*randPrior*) and (*pSherry*) [2] strategies, by the use of measurements to decide the effectiveness of all the strategies in order to find faults.

From the evaluation, the results showed that *HoceDanMafara* provides better results in the ten programs in term of APFD. Based on the measured performance obtained from the results, GA with reduced severity of fault prioritizes test cases more effective compared to using GA with the same severity of fault, random prioritization, and non-prioritize. The more the effectivess of regression testing strategies, the better the strategy, and the less cost of regression testing.

Although this paper has considerably achieved the intended goals as proposed by the strategy, there are many possible extensions that can be enhanced in the future. *HoceDanMafara* used existing tools in some of the phases while others are manually done. The prototype can be expanded to make it fully automated in the future so that all phases of the strategy can be integrated as a complete tool. The *HoceDanMafara* tool can be extended into a multi-language tool apart from Java, which can identify other OOP languages whose syntax is similar to that of Java, e.g. C++, C#. The newly tool extended can be applied to the tool by using existing concepts, which will help in having a broader opportunity and be more beneficial to a widespread of users.

References

[1] Rothermel, G., Roland, H.U., Chu, C., & Harrold, M.J. (2001). Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, 27(10), 929-948

[2] Purohit, G.N., & Sherry, A.M. (2014). Test suites prioritization for regression testing using genetic algorithm. *International Journal of Emerging Technologies in Computational and Applied Sciences.* 14(150). 255-259.

[3] You, L., & Lu, Y. (2012, May). A genetic algorithm for the time-aware regression testing reduction problem. In *Natural Computation (ICNC), 2012 Eighth International Conference on* (pp. 596-599). IEEE.

[4] Musa, S., Sultan, A. B. M., Abd-Ghani, A. A. B., & Baharom, S. (2014). A Regression Test Case Selection and Prioritization for Object-Oriented Programs using Dependency Graph and Genetic Algorithm, *Research Inventy: International Journal of Engineering And Science*, 4(7), 54-64.

[5] Musa, S., Sultan, A. B. M., Abd-Ghani, A. A. B., & Baharom, S. (19-20, 2018). Empirical Evaluation of Average Percentage of rate of Fault Detection of Software GA-based Regression Test Case Prioritization Strategy. *Proceeding of* International Conference on "Engineering & Technology, Computer, Basic & Applied Sciences"- ECBA-2018. Flora Creek Deluxe, Dubai, UAE.

[6] Shanmugam, R., & Uma G.V. (2012). Factors Oriented test case prioritization Strategy in Regression testing using Genetic Algorithm. *European Journal of Scientific Research*, 74(3), 389-402.

[7] Panigrahi, C. R., & Mall, R. (2014). A heuristic-based regression test case prioritization strategy for object-oriented programs. *Innovations in Systems and Software Engineering*, *10*(3), 155-163.

[8] Öztuna, D., Elhan, A. H., & Tüccar, E. (2006). Investigation of four different normality tests in terms of type 1 error rate and power under different distributions. *Turkish Journal of Medical Sciences*, *36*(3), 171-176.

[9] Lewsey, J. (2006). Medical statistics: a guide to data analysis and critical appraisal. *Annals of The Royal College of Surgeons of England*, *88*(6), 603.

[10] Thode, H. C. (2002). *Testing for normality* (Vol. 164). CRC pre