

## Comprehensive Cognizance of Regression Test Case Prioritization Techniques

Omdev Dahiya<sup>1</sup>, Kamna Solanki<sup>2</sup>

<sup>1,2</sup>University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak-124001, India

<sup>1</sup>Omdahiya21792@gmail.com, <sup>2</sup>Kamna.mdurohtak@gmail.com

### ABSTRACT

The worthiness of software can be effectively seen as it emerged as an integral asset that has discovered its application in each field ranging from cell phones to the medicinal treatment, covering practically all electronic devices on earth up to space, for example, satellite communication. In this way, the software has made a way of facilitating and changing the lives of humans. With such involvement of software in the life of humans, it must be developed in a sequential and well-defined way; else its quality will be compromised resulting in unreliable software. Software development is accomplished after undergoing different phases of SDLC. The quality of software can only be ensured by the means of testing. Many testing techniques are available to discover various errors/faults and debugging approaches are thus applied to remove them. One of the approaches for testing the software is exhaustive testing. But its limitation is that it can be applied only on very small programs. Due to time and cost constraints running the exhaustive testing on large programs is not a feasible idea. In regression testing, a whole test suite is executed again to make sure that changes made have not negatively affected the software. This paper has summarized the research findings of numerous researchers in the regression testing field.

**Key words:** Regression testing, Software, Software testing, Test case prioritization, Literature review.

### 1. INTRODUCTION

Software refers to a collection of related programs, its operating system procedure along with its documentation. It is developed by following a systematic approach defined in software engineering [1]. "Software engineering is an engineering discipline that is concerned with all aspects of software development" [1]. In the process of development of software, a lot of time is utilized in its testing, so it may be termed as the costliest phase. It is performed to ensure whether the developed software, meets quality standards [2]. The errors discovered in testing are therefore removed by different debugging techniques. One of the approaches

followed in testing is exhaustive testing. It states that the software program must be tested for all possible combinations of test cases to ensure that a software program becomes completely bug-free. Though, it has a limitation; it can be performed only on very small programs, running exhaustive testing of large programs is not a feasible idea considering the time, cost and resources involved.

### 2. REGRESSION TESTING

"Regression testing is selective retesting of a system or component to verify that modifications have not caused undesirable effects and that the system or component still complies with its specified requirements" [4] [5]. The test case prioritization approach increases test feasibility in the testing of software [3]. Software evolves with time, so the size of a software test suite also increases which often makes it costly to execute. Numerous researchers have shown that regression testing is a costly process and therefore it requires most of the collective expenditure of the software [6]. Yoo and Harman examined various regression testing approaches, to boost the implication of the accumulated test suite in regression testing. They elaborated regression testing approaches as- minimization, selection, and prioritization [7].

#### 2.1 Test Suite Minimization

In this approach, those test cases are eliminated, which turns out to be redundant about coverage of some set of program requirements and subsequently the number of test cases in a regression test suite is decreased [8]. According to it, from an economic point of view, only a subset of test suite may be executed. Minimization is at times called "test suite reduction" stating that the elimination of test cases is permanent [8].

#### 2.2 Test Case Selection

In this approach, those test cases are selected from the original test suite which focuses on testing the modified part of the software program. Instead of eliminating the test cases, test cases were filtered based on their ability for testing the modified source code portion [9].

### 2.3 Test Case Prioritization

According to this approach “execution of all the test cases is not feasible due to resource constraints, therefore the test cases are ordered such that those with higher priorities are run earlier than those with lower priorities based on some criterion” [10]. Prioritization techniques are usually preferred as no test cases are deleted from the initial test suite [11].

### 3. RELATED STUDIES

This section discusses the research work of different researchers in the field of regression testing for test suite optimization. The advantages and limitations of various test suite optimization techniques have been discussed in detail.

#### **Duggal and Suri, “Understanding regression testing techniques”, (2008).**

The authors have discussed regression testing techniques and their approaches in detail with search Algorithms for Test Case prioritization. In the paper, the authors elaborated on the whole organization of regression testing, enlightening the possible sphere of regression testing by discussing its scope and importance [12].

#### **Hareton and White, “Insights into regression testing”, (1989).**

The authors have discussed that regression testing is introduced as a way to measure the ease of retesting. They have further divided regression testing into corrective and progressive regression testing, whereas specification remains the same in the former approach while the latter one involves a customized specification. They have grouped the test cases into various classes and have shown their significance in the testing accordingly [13].

#### **Park *et al.* “Historical value-based approach for cost cognizant test case prioritization to improve the effectiveness of regression testing”, (2008).**

The authors proposed a technique based on the assessment of test cost and fault severities using past information. In their experiment, they have shown that their approach has yielded superior results in terms of APFDc (Cost based APFD), as compared to other techniques. The study conducted by authors has shown that using past information; fault severity and cost incurred can be approximated [14].

#### **Malishevsky *et al.* “Modelling cost-benefits tradeoffs for regression testing techniques”, (2002).**

The authors proposed various regression testing approaches to lessen the number of regression test cases that must be run to help testers in meetings their objectives quickly, as running the whole test suite again is not a feasible idea due to time and cost constraints. The authors have also

discussed the role of APFDc but it does not show whether it is cost-efficient or not. Thus cost-benefit models are proposed for cost-effectiveness to be used in case of test case Minimization, Selection, and prioritization; that capture earlier omitted factors and support cost-benefit analyses [15].

#### **Askarunisa *et al.* “Cost and Coverage Metrics for Measuring the Effectiveness of Test Case Prioritization Techniques”, (2010).**

The authors have discussed coverage and cost-based prioritization and have focused on the usefulness of prioritization techniques in terms of rate of fault detection and code coverage. Their study shows that the fault detection rate is improved by prioritization and they have proved that APFDc is a superior metric as compared to APFD. It helps in improving the fault detection rate by computing many metrics for different prioritization techniques that foretell their quality and efficiency. These metrics thus help in selecting a better prioritization technique which will help in reducing the time, cost and hence in the judicious use of resources during regression testing [16].

#### **Jacob and Ravi, “A novel approach for test suite prioritization”, (2014).**

The authors in this paper have discussed that for meeting performance goals, prioritization techniques will be very beneficial as they arrange test cases in a specific order thereby increasing their effectiveness. They have worked on prioritization using clustering. It is deduced that software quality is improved in the testing cycle when fault detection is increased with the use of APFD. This will be quite helpful, as executing an entire test suite is not feasible owing to limited resources, time and cost factor [17].

#### **Mayan and Ravi, “Structural software testing: A Hybrid algorithm for optimal test sequence selection during regression testing” (2015).**

The authors proposed a Hybrid Algorithm to overcome the disadvantage of genetic algorithm i.e there is less possibility to get the test cases required for testing the modified program when test cases are selected randomly. So, their algorithm selects the test cases based on, number of times the test case was capable to show its strength and these test cases have the potential to find maximum errors in no time. The selection of test cases varies from code coverage to time of their execution and high fault detection rate [18].

#### **Singh *et al.* “Systematic Literature Review on Regression Test Prioritization Techniques”, (2012).**

The author has discussed that despite various approaches followed by different techniques, the ultimate goal of test case prioritization remains the same, to increase the fault detection rate. The authors have stated that there is a need to carry out experimental comparisons amongst the prevailing techniques

as no broad techniques exist and new and effective techniques can be derived from them [19].

**Catal and Mishra, “Test case prioritization: a systematic mapping study”, (2012).**

The authors have focused on TCP studies by presenting an organized mapping in test case prioritization. Most of the reviews documented in their study were about the approval of diverse looks into TCP and the recommended solutions. The authors were able to extract many studies from the collected database which are interrelated with strong and weak points of several prioritization techniques [20].

**Kumar and Singh, “A Literature Survey on test case prioritization”, (2014).**

The authors have discussed coverage-based techniques and cost-effective techniques. They have concluded that the cost of development of software will ultimately be reduced if during the starting phases of the software development life cycle new techniques were implemented [21].

**Anderson *et al.* “Improving the Effectiveness of Test Suite through Mining Historical Data”, (2014).**

The authors have used test result history in developing two test prioritization techniques namely "most common failures" and "failures by association". The former approach depicts that the test cases which have failed in the past have a high probability to fail in the future. The latter approach works to improve the former one by using association rule mining. They have found both the techniques show the same performance in predicting future failures but works best when a small dataset is used instead of an entire historical dataset [22].

**Elbaum *et al.* “Techniques for Improving Regression Testing in Continuous Integration Development Environments”, (2014).**

The authors have presented their study on the cost-effectiveness of testing by test suite selection at the pre and post-submit testing stage. In the pre-submit stage, test suites are screened based on the logic that those test suites are executed again first; which have failed. The reason behind this is, recent failures are likely to predict future failures and some test suites which were skipped are also selected. In the post submit stage, the prioritization approach is followed instead of skipping test suites as done in the former approach, it is ensured that all test suites are executed while executing those first which tend to fail thereby shortening the feedback loop [23].

**Kiran *et al.* “A Literature Survey on TCP-Test Case Prioritization using the RT Regression Techniques”, (2015).**

The authors have discussed the “overall review of

regression testing with test case prioritization in detail. They also discussed various TCP techniques along with their strategies and algorithms developed till now. They have concluded that survey analysis eases the way of performing TCP. But, the benefits of the cost factors as well as of various parameters should be taken care of”. The testing of software consumes about 50% of the whole cost of its development [24].

**Solanki *et al.* “Test Case Prioritization: An Approach Based on Modified Ant Colony Optimization,” (2016).**

The authors have shown the usefulness of the regression testing technique in terms of APFD metrics. They have proposed a new algorithm for prioritizing the test cases in which for selecting their food source a different criterion is followed by natural ants. The algorithm has shown remarkable results when it was tested on various problems [25].

**Solanki *et al.* “A comparative evaluation of “m-ACO” technique for test suite prioritization”, (2016).**

The authors presented their work to show the comparative evaluation of their proposed algorithm against different TCP techniques using various case studies. The result obtained thus demonstrated the superiority of the algorithm. The application of the "m-aco" technique is quite significant in increasing the effectiveness and capability of a test suite [26].

**Labuschagne *et al.* “Measuring the Cost of Regression Testing in Practice”, (2017).**

The authors discussed the cost-benefit analysis of implementing regression testing. They analyzed that earlier studies have only emphasized the cost of maintaining the test suite. They studied java-based projects and have shown their findings [27].

**Alagoz *et al.* "A selection method for black-box regression testing with a statistically defined quality level", (2017).**

The authors proposed a selection method for black-box regression testing and have proved that efforts in regression testing can be minimized by clustering the test cases and consequently defined optimum clustering strategy in their theorem. These optimum clusters are identified and this tells us about the properties these clusters must have thereby ensuring minimum possible false positive probability. They have worked and demonstrated how to search for good clusters [28].

**Strandberg *et al.* “Regression Test Prioritization in a Nutshell”, (2017).**

The authors have worked on-suite builder tools which automatically determines the effective ordering of regression test cases. This has proved beneficial in removing various

shortcomings that are faced when regression testing is applied, such as time constraint in running all test suites, their prioritization and it is also checked whether some important test cases are not ignored [29].

**Aman *et al.* “A topic model and test history-based test case recommendation method for regression testing”, (2018).**

The authors have proposed a method for selecting test cases efficiently to make regression testing an effective one. A hybrid method was introduced which ensures no good test case is left and the effectiveness of the proposed method is shown by conducting regression testing of an industrial product [30].

**Agrawal and Kaur, “A comprehensive comparison of ant colony and hybrid particle swarm optimization algorithms through test case selection”, (2018).**

The authors have worked towards comprehensively comparing ACO and Hybrid PSO, through which they have demonstrated the importance of selecting the test cases [31].

**Dahiya and Solanki, “A systematic literature study of regression test case prioritization approaches”, (2018).**

The authors have conducted a systematic literature study using some research questions framed by them to select and analyze relevant research. After conducting the study, it is concluded that the current scope lies in developing a test case prioritization technique which is based on capturing the maximum number of user requirements [32].

**Haghighatkhah *et al.* “Test prioritization in continuous integration environments”, (2018).**

The authors have inspected a few open-source software projects and have worked towards catching regression faults earlier thus allowing developers to be flexible in dealing with changes [33].

**Khatibsyarbini *et al.* “Test case prioritization approach in regression testing: A systematic literature review”, (2018).**

The authors have performed an enhanced and deep review of TCP approaches in regression testing. After selecting studies via research questions and analyzing them according to different approaches proposed by researchers, they have shown the trends of publication in this field as well as the areas where improvements can be done [34].

**Mukherjee and Patnaik, “A survey on different approaches for software test case prioritization”, (2018).**

The authors have surveyed different approaches for test case prioritization. They have defined their search strategy and then studies were obtained which are thus filtered and fitted in the desired criterion. Selected studies are then

organized according to particular approaches. This will help both beginners as well as professionals [35].

**Chen *et al.* “Test case prioritization for object-oriented software: An adaptive random sequence approach based on clustering”, (2018).**

The authors have presented an approach for increasing the efficiency of regression testing for object-oriented software. The authors thus conducted experimental studies to prove the excellent results obtained in terms of early detection of faults [36].

**Ali *et al.* “On the search for industry-relevant regression testing research”, (2019).**

The authors have performed a systematic literature review of industrial level researches in regression testing so that research can be assessed and developed which addresses the industry needs. Their goal is to motivate and enable researchers to design and develop regression testing techniques that are industry-relevant and they are thus adopted in the industry [37].

**Dahiya *et al.* “Comparative Analysis of Regression Test Case Prioritization Techniques”, (2019).**

The authors have conducted a systematic comparative analysis based on the prioritization objectives framed by them. They have concluded that no two approaches can be compared based on performance in a certain environment as results may vary when factors and environments are changed [38].

So much of stress is laid on making software reliable and robust because there were many mishappenings in the past which has resulted in the loss of human life and property. Quality and reliability of software-intensive products are much more stressed when the software which is being developed is used in areas when human life is at stake such as in ventilators in hospitals, nuclear power plants and in a country's defense system. The following figure 1 illustrates the examples where the failure of a software system has cost millions of rupees as well as human life.



Img. Source - <https://www.cigniti.com/blog/top-10-mega-software-failures-of-2014/>

**Figure 1:** Picture depicting major software failures

There were many more examples where poor-quality software has resulted in a loss of money and human life.

- The “**Nest Smart Thermostat**” owned by **Google** was hit was a software glitch that left its customers literally out in cold. Nest clarified that it happened due to a Software Update in December 2015. [Root Cause: In-Effective Regression Testing]
- “**ObamaCare**”: Software swallows a third of insurance applications in the USA in 2013.
- “**Therac-25**” [35 Patients died due to Overdose of Radiations due to Regression Bug].
- “**British Airways - Disruption for 75,000 passengers**”. For the sixth time this year – a software failure led to significant delays on international flights and massive cancellations on local flights.
- “**Ariane 5**” [Space Shuttle by the U.S. collapsed just after its launch due to Regression Bug].

**Table I:** Advantages and Limitations of Various Approaches

Serial No.	Authors	Advantages	Limitations / Future Scope
1.	Duggal and Suri [2008]	Regression testing techniques and approaches were discussed [12].	No way is defined to select an appropriate approach for regression testing.
2.	Hareton and white [1989]	The study is conducted for similarities and differences between general testing and regression testing. Briefing of terminologies, strategies, and bifurcation of regression testing is performed [13].	The approach defined by the authors is only feasible if the size of the test data is reasonable. No way is defined to select the parts of the modified program for which retesting is to be done.
3.	Park <i>et al.</i> [2008]	The effectiveness of regression testing is improved by prioritizing the test cases through the approach proposed by the authors, in terms of APFDc [14].	Only two approaches were studied by the authors in their comparison. Technique on how to find the undetected defect in a test case is not provided, which may affect the test case prioritization.
4.	Malishevsky <i>et al.</i> [2002]	Cost-benefit models were proposed for regression testing approaches; that can detect factors that were missing before and supports the cost-benefit analyses which were not studied in the past [15].	No way is defined through which the models proposed by the authors can be implemented.
5.	Askarunisa <i>et al.</i> [2010]	The authors have shown that through prioritization, the fault detection rate is improved when the APFDc metric is used. The study has shown the superiority of the APFDc metric [16].	The effect of different cost factors was not incorporated while formulating the result and how the required changes will be made for implementing the prioritization technique is not discussed.

6.	Jacob and Ravi [2014]	The authors have performed the study to show the importance of test suite prioritization using clustering citing its advantages over running an entire test suite for testing [17].	No new approach is discussed.
7.	Mayan and Ravi [2015]	The authors have proposed the algorithm to overcome the drawbacks of various techniques in terms of code coverage within less time [18].	The proposed work shows the comparison of a few algorithms, its implementation on the different testing framework is required for depicting performance effectiveness in testing.
8.	Y. Singh <i>et al.</i> [2012]	The authors have concluded after studying hundreds of studies, that the ultimate goal of test case prioritization remains the same, to enhance the rate of fault detection [19].	As highlighted by the author also, there is a necessity to conduct experimental comparisons amongst the prevailing techniques so that new and effective techniques can be derived from them.
9.	Catal and Mishra [2012]	The authors performed their study on TCP techniques and they were able to reveal many strong and weak points of several prioritization techniques [20].	
10.	Kumar and Singh [2014]	The authors studied various techniques and argued upon reducing software development costs by using new techniques in the early phases of SDLC [21].	More studies could have been used to generalize the result achieved.
11.	Anderson <i>et al.</i> [2014]	The authors were able to develop new test case prioritization techniques by using test result history [22].	The proposed approach works best when a small dataset is used instead of an entire historical dataset.
12.	Elbaum <i>et al.</i> [2014]	The authors have presented techniques to make regression testing cost-effective [23].	The study could have been extended up to other datasets and many other limitations of regression test selection techniques could be revealed.
13.	Kiran <i>et al.</i> [2015]	The authors discussed the overall description of regression testing with test case prioritization with various TCP techniques along with their strategies and algorithms developed till now [24].	
14.	Solanki <i>et al.</i> [2016]	The authors have proposed a nature-inspired algorithm to be used in computer science for solving the optimization problem [25].	The proposed technique has not been implemented using industrial data and it also does not consider the cost of execution of test cases.
15.	Solanki <i>et al.</i> [2016]	The authors have shown the comparative evaluation of their proposed algorithm against different TCP techniques thus proving the superiority of their approach [26].	The proposed technique does not consider the severity of test cases and is not effective for a software system with a smaller test suite.
16.	Labuschagne <i>et al.</i> [2017]	The authors analyzed that earlier studies have only emphasized the cost of maintaining the test suite [27].	Work performed is limited to the study of java projects only.
17.	Alagoz <i>et al.</i> [2017]	The authors have defined a strategy to minimize efforts in regression testing [28].	There is no surety whether the clusters obtained are optimal.
18.	Strandberg <i>et al.</i> [2017]	The authors developed a suite builder tool that automatically determines the effective ordering of regression test cases [29].	There is a probability that certain test cases will be missed and further study is required to be done on how to be more systematic and optimized in the assignment of priority weights.
19.	Aman <i>et al.</i> [2018]	The authors developed a hybrid to ensure no useful test cases were left in test case selection [30].	Extension of this study can be done by introducing automation support to aid in finding the similarity in test cases.
20.	Agrawal and Kaur [2018]	The authors have worked towards comparing two metaheuristic algorithms showing the importance	The comparison of the algorithms can also be performed using different benchmark

		of selecting the test cases [31].	problems.
21.	Dahiya and Solanki [2018]	A systematic study is performed to find current trends and future scope in the field of regression testing [32].	
22.	Haghighatkah <i>et al.</i> [2018]	Authors have inspected software projects to guide developers to be flexible in dealing with changes [33].	As proposed by authors also, the study may be replicated on larger systems and in the industry as well.
23.	Khatibsyarbini <i>et al.</i> [2018]	The authors performed a systematic literature review to show current trends and future scope for test case prioritization techniques in regression testing [34].	
24.	Mukherjee and Patnaik [2018]	The authors have surveyed different approaches for test case prioritization. Studies are then selected and organized according to particular approaches [35].	
25.	Chen <i>et al.</i> [2018]	The authors presented an approach for making regression testing effective for object-oriented software and have proved their results via experimental study [36].	The sampling strategy can be improved.
26.	Ali <i>et al.</i> [2019]	The authors have performed a systematic literature review of industrial level researches in regression testing so that research can be assessed and developed which addresses the industry needs [37].	
27.	Dahiya <i>et al.</i> [2019]	The authors have conducted a systematic comparative analysis of test case prioritization techniques in regression testing based on the prioritization objectives framed by them [38].	An experimental study can be performed to add more weight to the research.

#### 4. CONCLUSION

Regression testing is a testing activity done to retest the software for the modifications it has undergone. It is a time consuming and costly process, so various researchers have done their work to make it an effective process. This paper has presented the work of numerous authors on test case prioritization approaches. The researchers have developed various techniques for test suite optimization. They have developed algorithms for defining the sequence of test suite execution, to achieve remarkable results. Though a lot of work is done in making the regression testing process efficient, there is future scope, which can be achieved by combining various approaches and consecutively achieving new and better results.

#### REFERENCES

1. Sommerville, Ian **“What is software engineering?”**. Software Engineering (8th ed.). Harlow, England: Pearson Education p.p. 7, 2007.
2. G. J. Myers, T. M. Thomas, and C. Sandler, **The Art of Software Testing**, vol. 1. John Wiley & Sons, 2004.
3. G. Rothermel, R. H. Untch, C. C. Chu, and M. J. Harrold, **“Test case prioritization: an empirical**

**study”**, in Software Maintenance. (ICSM '99) *Proceedings. IEEE International Conference on software maintenance*, pp.179–188, 1999. <https://doi.org/10.1109/ICSM.1999.792604>

4. H. K. N. Leung, **“Insights into Regression Testing,”** *Proceedings of the International Conference on Software Maintenance*, pp. 60–69, 1989.
5. S. Elbaum, G. Rothermel, and J. Penix, **“Techniques for improving regression testing in continuous integration development environments”**, in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering -* pp. 235–245, 2014. <https://doi.org/10.1145/2635868.2635910>
6. P. K. Chittimalli and M. J. Harrold, **“Recomputing coverage information to assist regression testing,”** *IEEE Transactions on Software Engineering*, vol. 35, no. 4, pp. 452–469, 2009.
7. S. Yoo and M. Harman, **“Regression Testing Minimisation, Selection, and Prioritisation: A Survey”**, *Software Testing, Verification and Reliability*, 22(2), pp.67-120, 2007. <https://doi.org/10.1002/stv.430>
8. D. Jeffrey and N. Gupta, **“Improving fault detection capability by selectively retaining test cases during test suite reduction,”** *IEEE Transactions on Software Engineering*, vol. 33, no. 2, pp. 108–123, Feb. 2007.

- <https://doi.org/10.1109/TSE.2007.18>
9. S. Elbaum, P. Kallakuri, A. G. Malishevsky, G. Rothermel, and S. Kanduri, “**Understanding the effects of changes on the cost-effectiveness of regression testing techniques**,” *Journal of Software Testing, Verification and Reliability*, vol. 12, no. 2, pp. 65–83, 2003.  
<https://doi.org/10.1002/stvr.263>
  10. S. Elbaum, A. G. Malishevsky, and G. Rothermel, “**Test case prioritization: a family of empirical studies**,” *IEEE Transactions on Software Engineering*, vol. 28, no. 2, pp. 159–182, 2002.  
<https://doi.org/10.1109/32.988497>
  11. S. Raju, G.V. Uma, “**Factors oriented test case prioritization technique in regression testing using genetic algorithm**”, *European Journal of Scientific Research*, 74(3), pp.389-402, 2012.
  12. G. Duggal and B. Suri, “**Understanding regression testing techniques**”, *Proceedings of 2nd National Conference on Challenges and Opportunities in Information Technology*, (2008).
  13. Hareton and Lee White, “**Insights into regression testing**”, *Proceedings of the Conference on Software Maintenance*, pp 60-69, October 1989.
  14. H. Park, H. Ryu, J. Baik “**Historical value-based approach for cost cognizant test case prioritization to improve the effectiveness of regression testing**”, In *Secure System Integration and Reliability Improvement*, pp. 39-46. IEEE, 2008.  
<https://doi.org/10.1109/SSIRI.2008.52>
  15. A. G. Malishevsky, G. Rothermel, and S. Elbaum, “**Modelling cost-benefits tradeoffs for regression testing techniques**”, In *Proceedings. International Conference on*, pp. 204-213. IEEE, 2002.
  16. A. Askarunisa, L. Shanmugapriya, N. Ramaraj, “**Cost and Coverage Metrics for Measuring the Effectiveness of Test Case Prioritization Techniques**”, *INFOCOMP*, 9(1), pp.43-52, 2010.
  17. T. P. Jacob and T. A. Ravi, “**A novel approach for test suite prioritization**”, 2014.  
<https://doi.org/10.3844/jcssp.2014.138.142>
  18. J. A. Mayan and T. Ravi, “**Structural software testing: A Hybrid algorithm for optimal test sequence selection during regression testing**”, *International Journal of Engineering and Technology*, 7(1), pp.270-9. 2015.
  19. Y. Singh, “**Systematic Literature Review on Regression Test Prioritization Techniques**” *Informatica* 36, no. 4, 2012.
  20. Catal and D. Mishra, “**Test case prioritization: a systematic mapping study**,” *Software Quality Journal*, vol. 21, no. 3, pp. 445–478, 2012.  
<https://doi.org/10.1007/s11219-012-9181-z>
  21. A. Kumar and K. Singh, “**A Literature Survey on test case prioritization**,” *CompuSoft*, 2014.
  22. J. Anderson, S. Salem, H. Do, “**Improving the Effectiveness of Test Suite through Mining Historical Data**”, In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 142-151. ACM, 2014.  
<https://doi.org/10.1145/2597073.2597084>
  23. S. Elbaum, G. Rothermel, J. Penix, “**Techniques for Improving Regression Testing in Continuous Integration Development Environments**”, In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 235-245. ACM, 2014.  
<https://doi.org/10.1145/2635868.2635910>
  24. P. Kiran and K. Chandraprakash, “**A Literature Survey on TCP-Test Case Prioritization using the RT- Regression Techniques**” *Global Journal of research in engineering*, 2015.
  25. K. Solanki, Y. Singh, S. Dalal, and P. Srivastava, “**Test Case Prioritization: An Approach Based on Modified Ant Colony Optimization**,” In *Emerging Research in Computing, Information, Communication, and Applications*, pp. 213-223. Springer, 2016.
  26. K. Solanki, Y. Singh, S. Dalal, “**A Comparative Evaluation of “m-ACO” Technique for Test Suite Prioritization**”, *Indian Journal of Science and Technology*, Vol 9(30), 2016.
  27. A. Labuschagne, L. Inozemtseva, R. Holmes. “**Measuring the Cost of Regression Testing in Practice- A Study of Java Projects Using Continuous Integration**”, 2017.
  28. I. Alagoz, T. Herpel, R. German, "A selection method for black-box regression testing with a statistically defined quality level", In *Software Testing, Verification and Validation (ICST)*, *IEEE International Conference on*, pp. 114-125. IEEE, 2017.  
<https://doi.org/10.1109/ICST.2017.18>
  29. P. Strandberg, W. Afzal, T. Ostrand, E. Weyuker and D. Sundmark, “**Automated system level Regression Test Prioritization in a Nutshell**”, *IEEE Software*. 2017.
  30. Aman, H., Nakano, T., Ogasawara, H. and Kawahara, M., “**A topic model and test history-based test case recommendation method for regression testing**”. In *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 392-397). April 2018.
  31. A.P. Agrawal and A. Kaur, “**A comprehensive comparison of ant colony and hybrid particle swarm optimization algorithms through test case selection**”, In *Data engineering and intelligent computing* (pp. 397-405). Springer, Singapore, 2018.
  32. O. Dahiya and K. Solanki, “**A systematic literature study of regression test case prioritization approaches**”, *International Journal of Engineering & Technology*, Vol. 7, No. 4, pp.2184-2191, 2018.  
<https://doi.org/10.14419/ijet.v7i4.15805>
  33. A. Haghghatkhah, M. Mäntylä, M. Oivo and P. Kuvaja, “**Test prioritization in continuous integration environments**”. *Journal of Systems and Software*, 146, pp.80-98, 2018.



34. M. Khatibsyarbini, M.A. Isa, D.N. Jawawi, and R. Tumeng, “**Test case prioritization approaches in regression testing: A systematic literature review**”, *Information and Software Technology*, 93, pp.74-93, 2018.
35. R. Mukherjee and K.S. Patnaik, “**A survey on different approaches for software test case prioritization**”. *Journal of King Saud University-Computer and Information Sciences.*, 2018.  
<https://doi.org/10.1016/j.jksuci.2018.09.005>
36. J. Chen, L. Zhu, T.Y. Chen, D. Towey, F.C. Kuo, R. Huang and Y. Guo, “**Test case prioritization for object-oriented software: An adaptive random sequence approach based on clustering**”. *Journal of Systems and Software*, 135, pp.107-125, 2018.
37. N. bin Ali, E. Engström, M. Taromirad, M.R. Mousavi, N.M. Minhas, D. Helgesson, S. Kunze and M. Varshosaz, “**On the search for industry-relevant regression testing research**”, *Empirical Software Engineering*, pp.1-36, 2019.
38. O. Dahiya, K. Solanki and S. dalal, “**Comparative Analysis of Regression Test Case Prioritization Techniques**”, *International journal of advanced trends in computer science and engineering*, Volume 8 No. 4, pp. 1521-1531, 2019.  
<https://doi.org/10.30534/ijatcse/2019/74842019>