

*Research Article*

# A Novel Approach for Efficient Selection of Test Case Prioritization Techniques

Mrs. Sarika Chopdhe and Prof. Pooja Patil

Department of Computer Engineering, D Y Patil Clg of engineering, Ambi

Received 10 Nov 2020, Accepted 10 Dec 2020, Available online 01 Feb 2021, **Special Issue-8 (Feb 2021)**

## Abstract

*Test case prioritization techniques schedule test cases to lessen the cost of regression checking out and to maximize a few objective function. Test cases are prioritized such that those check cases which are more important under some standards are executed in advance in regression checking out procedure. Test case prioritization allows to improve the effectiveness of Regression Testing. However, running all the take a look at cases in the test suite is prohibitive in most instances. Optimization of test case execution time to maximize the early fault detection rate of the original test suite will assist to decrease the rate of regression testing. Because of the resource and time constraint, it turns into necessary to broaden some strategies which help to minimize existing take a look at suites by disposing of redundant test instances and prioritizing them. We make the preparation and test information via robotized change of User interface segments taken from the page wireframes. In light of the plan rules and the test varieties, we present potential defects in direct connection to the structure input. In light of the preparation information and the multifaceted nature of our situations, various models, for example, Reinforcement Learning and ANN can be picked. When the model is concluded, we can prepare our model to catch UI absconds. Along these lines, we can accomplish more prominent gauges as far as upgraded Test advancement inclusion of UI segments. The main purpose in the back of is the absence of hints for the selection of TCP strategies. Hence, this piece of research introduces a novel technique for class of Test case prioritization strategies the usage of fuzzy common sense to assist the efficient selection of test case prioritization strategies. This work is an expansion of as of now proposed determination outline for experiment prioritization procedures.*

**Keywords:** Test case prioritization (TCP), User Interface, Regression testing, fuzzy method, Optimization.

## Introduction

The expression "test prioritization" refers to the subjective and complex part of trying out that permits testers to manipulate risks, plan tests, bear in mind cost value, and be analytical about which take a look at to run in the context of the particular project. This technique is well referred to as Test Case Prioritization, which is a method of prioritizing and scheduling check instances. This procedure is used with a purpose to run check instances of better priority with a purpose to decrease time, cost and efforts for the duration of software program trying out phase. Furthermore, check case prioritization affords help with regression trying out and improves its performance. Through this method, testers can without difficulty run check instances, which have the highest priority and provide in advance defect faults. Also, an improved rate of fault detection for the duration of the trying out phase, lets in faster remarks of the system that is under check. With test case prioritization software engineers can get assistance in correcting faults earlier than might in any

other case be possible. Moreover, to determine the concern of take a look at instances, various factors depending upon the want of the software are decided. In testing part Regression testing helps in protection section is surely the manner of retesting the up to date software program to ensure that new errors have no longer been delivered into in advance validated code. Moreover, regression checking out should absorb as little time as possible in the feel of executing as few check instances as possible. Due to its high-priced nature various techniques exist in literature that concentrate on price. These are: Re-execute all; Test case Minimization/ Reduction; Test Case Selection; Test Case Prioritization; Hybrid approach. Today, businesses have better gadget getting to know algorithms for pattern analysis and processing large volumes of records that result in better run-time selections. For instance, in the course of a software program upgrade, system getting to know algorithms can traverse the code to discover key adjustments in functionality and hyperlink them to the requirements which will identify take a look at cases. This enables

optimize testing and prevents the making of decisions on warm spots that would result in failure. We are pioneering a strategy that applies AI to QA beyond take a look at optimization and prediction of failure to prevention through self-reliant technologies. This method would lead to a huge discount in standard trying out effort and also lessen dependency on humans. Its built-in intelligence and self-learning abilities make it a superior answer for identifying important paths for testing and for growing checking out accuracy.

This paper centers around experiment prioritization procedures. Presently days analyzers may wish to expand the inclusion of code in the product under test at a quicker rate, increment or improve their trust in the dependability of the product in lesser time, or increment the rate at which test suites distinguishes blames in that framework during relapse testing. The significant issues of code-based prioritization strategies is that they center just around number of issues recognized and, subsequently, treats all issues similarly.

## B. Motivation

Requirement mainly based on test case prioritization cope with such troubles by means of assigning the priority to test instances on requirement coverage based totally. Major Issue with specification based and requirement based totally Test Case Prioritization is that it is not well organized and less efficient.

## C. Objectives

- To provide efficiency in the selection of the test-case.
- To optimize the array or collection of test-cases.
- To create the method more reliable.
- Software developers and testers intend to enhance the rate of error detection.
- To improve the mechanism working of the prioritizing test cases according to fuzzy logic rule.

## Review of Literature

In this paper, the authors proposed a brand new check case prioritization approach using Genetic Algorithm. This test analyzes the genetic algorithm with regard to effectiveness and time overhead by utilizing structurally-based criterion to prioritize test instances. An Average Percentage of Faults Detected metric is used to decide the effectiveness of the new take a look at case orderings. Experimental analysis demonstrates that their approach can have This paper identifies and evaluates the challenges associated with time-aware prioritization. [1] In this paper, the author proposed a set of rules which is primarily based on analysis of the share of test instances accomplished to discover the faults and on APFD metric's outcomes. Outcomes reveal that their algorithms can also achieve better

execution. For instance, within the first project if most effective take a look at cases might be melt down due to resource constraint, random method could find greater or much less fault rates; while our proposed set of rules detects about more fault rates as compared to existing. They had additionally demonstrated their consequences with the aid of widespread APFD metric. [2]

In this paper, the author proposes a 3-phase technique to solve check case prioritization. In the first phase, we are removing redundant check cases with the aid of easy matrix operation. In the second one phase, test cases are decided on from the check suite such that decided on check instances constitute the minimal set which covers all faults and additionally at the minimum execution time. For this phase, we are the usage of multi goal particle swarm optimization which optimizes fault insurance and execution time. In the third phase, we allocate priority to check cases received from the second one phase. Priority is received via calculating the ratio of fault coverage to the execution time of test instances. [3]

In this paper, the author focuses on finding a way which gives a measure of the attempt to be spent on the trying out phase. This paper provides attempt estimates during pre-coding and put up-coding phases using neural community to predict extra accurately. The proposed effort estimation models for pre-coding phase primarily based on use case factor and soft computing technique- neural community has been applied to enhance upon the accuracy. The approach that has been accompanied and the metric proposed have an advantage that it produces correct results. For the submit coding effort estimation the proposed version expected the attempt based on and used neural network to enhance upon accuracy and the effects were found to show that the proposed estimation is in synchronization with the traditional attempt estimation fashions. [4]

In this paper, the author developed a brand new hybrid optimization technique called Hybrid Evolutionary Firefly Algorithm is proposed. The method combines the same old Firefly Algorithm with the evolutionary operations of Differential Evolution method to enhance the searching accuracy and facts sharing among the fireflies. The Hybrid Evolutionary Firefly Algorithm approach is used to estimate the parameters in a complex and nonlinear biological model to deal with its effectiveness in high dimensional and nonlinear trouble. This is crucial to make sure that the speed performance will now not be suffering from the hassle complexity. Secondly, the course of the fireflies can be introduced to the technique so that the firefly movements can be improved substantially. Lastly, the proposed method should be tested to estimate the parameters in more complicated problems which include noise and identifiable. [5]

This paper affords a novel technique for take a look at case prioritization in the course of regression testing of applications which have assertions using fuzzy logic.

The most important goal of this technique is to prioritize the check instances according to their envisioned potential in violating a given application assertion. To increase the proposed technique, we make use of fuzzy logic strategies to estimate the effectiveness of a given test case in violating an assertion based totally at the records of the test cases in preceding checking out operations. [6]

This paper provides an method to prioritize regression take a look at cases based on three elements which are charge of fault detection [6], percent of fault detected and risk detection ability. The proposed method is as compared with one-of-a-kind prioritization techniques which includes no prioritization, reverse prioritization, random prioritization. [7]

In this paper we've got presented the various types of regression checking out strategies their classifications provided by diverse researchers , explaining selective and prioritizing test cases for regression trying out in detail. Retest all method is one of the conventional methods for regression testing in which all of the tests in the existing check suite are rerun. So the retest all technique is very high priced compared to strategies if you want to be discussed in addition as regression take a look at suites are pricey to execute in full because it require more time and budget. [8]

The goal of this research is to propose "prioritizing factors" that better mirror the real-world scenario for take a look at case prioritization inside the specification-primarily based environment: (1) requirement severity score and inter-case dependency, and to optimize the take a look at case arrangement through the software of meta-heuristics. The inter-case dependency may be formulated as a sequential ordering trouble (SOP), a NP-complete trouble for which the priority courting exists. [9]

This paper gives a machine-level, cost-driven approach to check case prioritization referred to as the Prioritization of Requirements for Test (PORT). PORT involves analysing and assigning cost to every requirement the use of the following four elements: requirements volatility, consumer precedence, implementation complexity, and fault proneness. System take a look at cases are prioritized such that the test cases for necessities with better precedence are executed in advance during device test. [10]

### Proposed Methodology

This piece of research is an extension of proposed selection schema and provides a Model for selection of test case prioritization technique based on three factors: (i) requirement coverage, (ii) efforts and, (iii) complexity According to survey performed complexity can be taken on the scale of 1-10 usually defined by the developer and analyst.. effort is measure in terms of two in two terms: Final Total effort (FTE) and Average Effort (AE). relevant project attributes/features is done to identify TCP techniques covering maximum project attributes consequently requirements. Different steps of proposed approach are:

- Identifying project features' in terms of relevance and hence coverage of requirements.
- Identify the complexity of testing techniques.
- calculating testing effort.
- classify TCP techniques using fuzzy inference.

Advantages:

- 1.Reduce time cost to select test case.
  - 2.Enhance prioritization technique. Modules:
- In this different modules are used:

Module 1:

Tester: Tester will first register to the system and then login to the system with valid username and password. Then tester will enter the number of test cases and according to the test case prioritization method tester will get the result whether the test cases will get selected or discard.

Module 2:

Admin: Admin will login to the system by using valid username and password. Admin can view the details of the tester. Admin will only view the how many test cases will get succeed and how many get rejected. Overall, admin will manage the records of the tester for the improvement of the work.

A. Architecture

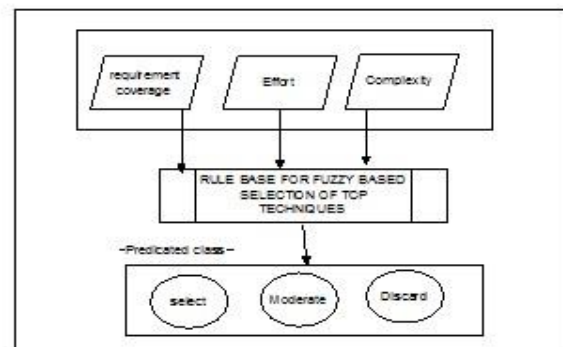


Fig. 1. Proposed System Architecture

Explanation:

In this framework, first we are login to the framework by utilizing OTP. After login to the framework we are going to view our profile, Then we can test the various test cases according to the complexity of that test cases that the test cases will be complex or relevant, after entering various test cases we are getting the result as that test cases will be selected or discarded. At last that test cases result will be get as the conclusive outcome from the framework.

B. Algorithms

Hybrid Approach:

In this approach there are three factors: requirement coverage, efforts, complexity, and time. Input:

- Relevance of selected Test Case Prioritization Techniques based on maximum requirement coverage.
- Complexity of selected Test Case Prioritization techniques

- Average Effort
- Time to execute Output:
- Final class : Test Case Prioritization Techniques.

Begin

1. Identify input variables (linguistic variables) i.e. relevance, Average Effort, Complexity (i.e. initialization)
2. Mapping of fuzzy sets to input variables by constructing the membership functions (i.e. initialization)
3. Formation of rules to create the rule base (i.e. initialization)
4. Conversion of input data (i.e. fuzzification)
5. Assessment of available rules in the rule base (i.e. inference)
6. Merge all the results achieved from available rules (inference)
7. Mapping of output data (i.e. defuzzification).

End

### C. Mathematical Model

#### 1. Mathematical equation:

The algorithm implemented in this project is described as:

Size = Count of Test Case

Effort = Size / No. of test cases to be executed

Time = Size X Time Required (per person (Hour))

Cost = Effort X Cost (of per day)

$$P = 1 - \frac{Tf_1 + Tf_2 + \dots + Tf_m}{mn} + \frac{1}{2n} \quad (1)$$

where,

- T : Test case under evaluation
- m : number of faults contained in the test cases
- n : total no. of test cases
- Tfi : position of first test in T that exposes fault i.

### Result and Discussion

Experiments are done by a personal computer with a configuration: Intel (R) Core (TM) i5-6700HQ CPU @ 2.60GHz, 16GB memory, Windows 7, MySQL Server 5.1 and Jdk 1.8. In our system, We compared the proposed disease prediction accuracy on number of samples and show the result graphically. Let see the following graph and table shows the disease prediction accuracy result based on classification technique. The overall accuracy of proposed technique is enhanced as compared to existing techniques. So our proposed system accuracy is better than existing system. So this works gives better test case prioritization results as compare to existing method.

#### A. Results and Performance

Number of Test Case Prioritization selection:

S.No	Class	No. of Test Prioritization Case techniques
1	Select	10
2	Discard	7
3	Moderate	5

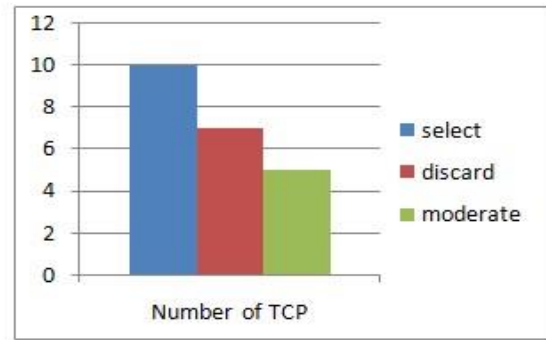


Fig. 2. Number of test case prioritization selection in different classes according to fuzzy based method

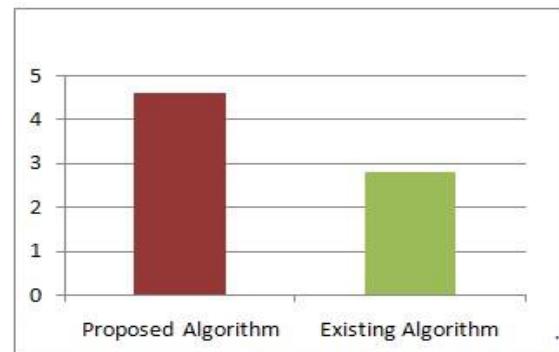


Fig. 3. Algorithms Comparison

### Conclusion

The objectives for the research have been met. There is a bunch of example approaches using machine learning presented from the different aspects of the software testing. These researches answer the question how AI implementations could serve the testing. In the beginning of starting this research it was unclear what would the findings be and what kind of document would this be in the end. The scope and data in the document prove that the field of AI testing has grown rapidly during the last few years and there actually was much more information available that was assumed in the beginning. At present the framework in this examination we have proposed a novel established system for the association of Test case prioritization methods utilizing Fuzzy Logic, for the variety of experiment prioritization strategy dependent on three components: necessity inclusion, endeavors and, intricacy. This exertion is an adjournment of recently proposed combination composition for experiment prioritization strategies. Prioritizing check instances will limit the time, attempt and value of testing and uncover maximum faults inside the software. The current days of manual and automated UI testing gradually become ineffective when contrasted with an MLbased solution. Deployment of this work would unquestionably prove to be extraordinary decline in the testers' time and effort complexity.

### Acknowledgment

The authors would like to thank the researchers as well as publishers for making their resources available and

for their guidance. We are thankful to the authorities of University of Pune and concern members of CPGCON 2020 conference, organized by, for their constant guidelines and support. We are also thankful to the reviewer for their valuable suggestions. We also thank the college authorities for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

## References

- [1] R.Krishnamoorthi and S.A.Sahaaya Arul Mary, "Regression Test Suite Prioritization using Genetic Algorithms", *International Journal of Hybrid Information Technology* Vol.2, No.3, July, 2009.
- [2] Thillaikarasi Muthusamy and Dr. Seetharaman.K, "Effectiveness of test case prioritization technique based on Regression Testing", *International Journal of Software Engineering Applications (IJSEA)*, Vol.5, No.6, November 2014.
- [3] Manika Tyagi and Sona Malhotra "Test Case Prioritization using Multi Objective Particle Swarm Optimizer" 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT).
- [4] Chintala Abhishek, Veginati Pavan Kumar, Harish Vitta, Praveen Ranjan Srivastava "Test Effort Estimation Using Neural Network", *J. Software Engineering Applications*, 2010.
- [5] Afnizanfaizal Abdullah, Safaai Deris, Mohd Saberi Mohamad, and Siti Zaiton Mohd Hashim "A New Hybrid Firefly Algorithm for Complex and Nonlinear Problem", *Distributed Computing and Artificial Intelligence*, pp. 673-680, Springer-Verlag Berlin Heidelberg 2012.
- [6] Ali M. Alakeel, "Using Fuzzy Logic in Test Case Prioritization for Regression Testing Programs with Assertions," *Scientific World J*, vol2014, Article ID-316014, 2014.
- [7] D. Silva, R. Rabelo, M. Campanha, P. S. Neto, P. A. Oliveira and R. Britto, "A hybrid approach for test case prioritization and selection," *IEEE Congress on Evolutionary Computation (CEC)*, pp. 4508-4515, July 2016.
- [8] Tyagi, M.; Malhotra, S.: An approach for test case prioritization based on three factors. *Int. J. Inf. Technol. Comput. Sci.* 4, 79-86 (2015)
- [9] G.Duggal, B.Suri, "Understanding Regression Testing Techniques", COIT, 2008.
- [10] Varun kumar, Sujata, Mohit Kumar, "Test case prioritization using fault severity", *International Journal of Computer Science and Technology (IJCST)*, Vol-1, Issue-1, pp-67-71, 2010.