International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) ISSN: 2395-5325
Volume 3, Special Issue 3 (September '2017)
Proceeding of - International Conference on Recent Trends in Computational Life Science and Information Technology,
Conference held at Tiruppur Kumaran College for Women,Tirupur,Tamilnadu, India.

AN OVERVIEW ON TEST CASE REDUCTION METHODS FOR DATA MINING TECHNIQUES

D.Yamuna,
M.Phil., Research Scholar,
PG & Research Department of Computer Science,
Vidyasagar College of Arts and Science,
Udumalpet, Tamilnadu,India.

Dr.N.Sasirekha,
Associate Professor,
PG & Research Department of Computer Science,
Vidyasagar College of Arts and Science,
Udumalpet, Tamilnadu,India.

Abstract: Software testing is some action meant by evaluating an attribute and determining that program or system meets its required results. This is significant accomplishment in software development. Test case selection is a critical action in testing because the number of automatically generated test cases is regularly huge and probably unfeasible. As well, a large number of test cases are unnecessary. It trains similar features of the application and they are capable of uncovering a similar set of faults. Data mining finds similar patterns in test cases which helped us in finding out redundancy incorporated by automatic generated test cases. We proposed a methodology based on data mining by which we can significantly reduce the test suite. The paper aimed to selecting the fewer related test cases at the same time as providing the best possible model from which test cases are generated by using data mining techniques.

Keywords: Software Testing, Data Mining, Test Case Reduction, Evaluation Parameters, Test Suite.

I.INTRODUCTION

Software testing is the major challenge of complex software systems. In this field, accuracy and running time are two major performance factors frequently employed to evaluate the reliability of software testing. It is mostly done by re-running existing test cases against the modified code to determine whether the changes affect anything. This requires a lot of cost and time, which increases as the size and the complexity of the software increases. Instead of re-running all the test cases, a number of different approaches were studied to solve testing problems. There has been an explosion in the use of data mining techniques in the exploration and analysis of large quantities of data in order to discover meaningful patterns and rules[1**].**

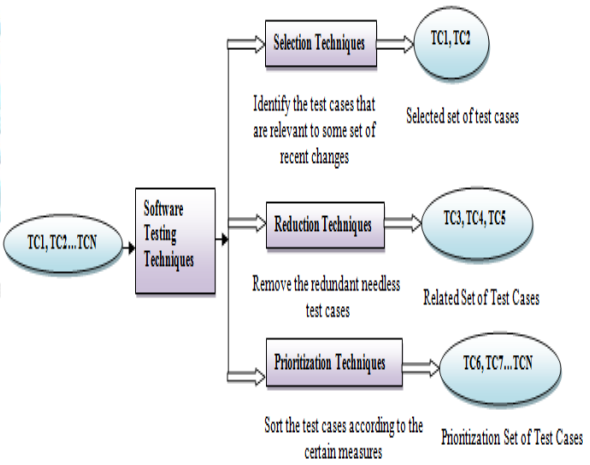

Figure 1: Testing Approaches

Data mining models were introduced for software testing to design a minimal set of test cases. This helps solving testing problems with large scale systems that are usually accompanied by thousands set of test cases, where it is considered impossible to re-run all of them each time a system update is applied. Therefore, data mining is investigated to handle such cases. In this paper, we investigate the different techniques proposed to solve the regression testing problems, where a comprehensive study is conducted for analysis [1]. The main concerns of these techniques when conducting software tests are adequate coverage, early discovery of bugs and time related constraints [1]. The main purpose of test case reduction is to decrease the number of test cases in order to minimize the time and cost of executing them. This paper used the data mining approach, mainly because of its ability to extract patterns of test cases that are invisible.

II.LITERATURE SURVEY

Zhenyu chen, baowen xu, xiaofang zhang and changhai nie Proposed the Requirement Based algorithm. In this algorithm, Subsets of test cases that fulfill the requirements are chosen. Model-checker take a finite state model and temporal logic property as input and as a result counter example will be returned if the property is no not fulfilled.The reduction is significant but one drawback is the run time complexity [16].

Siripong Roongruangsuwan and Jirapun Daengdej used the Coverage Based technique. It use an artificial intelligent concept of case based reasoning (CBR).propose three methods using CBR: Test Case Complexity for Filtering (TCCF), Test Case Impact for Filtering (TCIF) and Path Coverage for Filtering (PCF) Method.In PCF, the number of test cases is minimized more than other algorithms and it consumes the least reduction time [17].

B.subashini, d.jeyamala proposed the Clustering technique. It uses data mining approach of clustering technique to reduce the test suite. By using clustering the program can be checked with any one of the clustered test cases rather than with the entire test case that is produced by the independent

paths. The number of test cases is reduced and the efficiency of software testing is improved [3].

Haider, A.A.; Rafiq, S.; Nadeem used Fuzzy logic to an expert system that use a technique and level of testing based on a defined objective function, similar to human judgment using fuzzy logic based classification [18].

III.SOFTWARE TESTING

Software testing is the process of validation and verification of the software product. Effective software testing will contribute to the delivery of reliable and quality oriented software product, more satisfied users, lower maintenance cost, and more accurate and reliable result. However, ineffective testing will lead to the opposite results; low quality products, unhappy users, increased maintenance costs, unreliable and inaccurate results. Hence, software testing is a necessary and important activity of software development process [2].

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. The mechanism for determining whether a software program or system has passed or failed such a test is known as software testing. It may take many test cases to determine that a software program or system is functioning correctly. There are different kinds of software testing techniques. Broadly, there are two basic types of testing techniques: Black box testing and White box testing. Black Box Testing is also called functional testing because this testing is only concerned with the functionality of the software being developed. Reducing cost is the target of researchers on the use of test suite reduction techniques. Therefore, a number of different methods have been studied to deal with test suits such as minimization, selection and prioritization. Test suite minimization or reduction aims to reduce the number of tests to run.

IV. DATA MINING

Data mining [4, 5] is a semi automated process of finding patterns in the data. It is basically knowledge discovery in data. This knowledge discovered can be represented by a set of rules, equations relating different variables and other mechanisms of predicting outcomes. The manual component of data mining [4] is the preprocessing phase where data is prepared acceptable by the algorithms and post processing phase involving discovering patterns to find out new ones that are useful. There are three main techniques in data mining classification, association rules and clustering [4].

- Classification is a technique that classifies data into different classes by building models like decision trees. By using these models it predicts the behavior of future data.
- Association rules are the techniques used to find relationships or associations between different entities of an instance. With these associations we can predict the nature of one when the other changes [6].
- Clustering [7] is a technique used in finding clusters of points in the given data. In other words clustering is grouping together similar points into a single cluster. This behavior of grouping can be found out by different metrics like distance, density and grid based approaches. Within a cluster all set of points in that cluster are found to have similar behavior. In order to ease this process of data clustering, in the next section we introduce a tool called weka [8] which helps us in filling the gap between the processes of software testing and knowledge mining.

Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites. The test suite optimization process involves generation of effective test cases in a test suite that can cover the given system under test within less time. In the proposed approach, the test cases are selected by data mining techniques. Now, the approach generates a few efficient test cases that can cover the model within less amount of time [3].

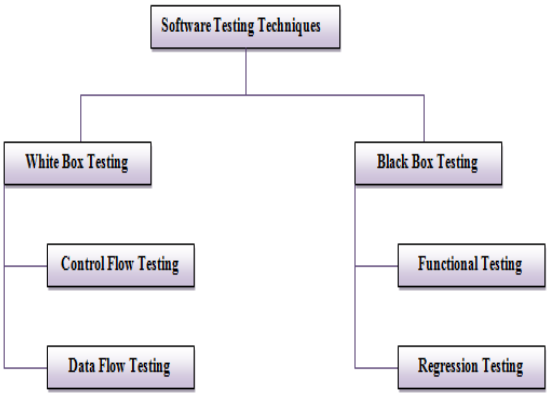

Figure 2: Basic Types of Software testing

V.TEST CASES REDUCTION TECHNIQUES

Test cases reduction techniques try to remove redundant test cases of a test suite. The test suite minimization problem can be formally stated as follows. Given:

A test suite T of test cases {t1, t2, t3. . . tk}.

- A set of testing requirements {r1, r2, r3. . . rn} that must be satisfied to provide the desired testing coverage of the program.
- Subsets {T1, T2, T3, . . ., Tn} of T, one associated with each of the ri's, such that any one of the test cases tjs belonging to Ti satisfies ri[9].

a) Selection Evaluation Parameters for software Testing

Several parameters have been used throughout the recent research to evaluate the different regression testing techniques. Parameters have varied depending on whether the technique applies selection, reduction or prioritization. Those parameters represent a set of basis in which selective techniques can be compared and evaluated.

- **Inclusiveness:** This parameter measures the extent to which a selective re-test strategy S selects modification-revealing tests from the initial test suit T for inclusion in T′, where a test $T_i \in T$ is a

modification revealing if it produces different outputs in P and P[10].

- **Efficiency:** This parameter measures the efficiency of the selection algorithm in terms of space and time requirements. Space efficiency is affected by the test history and program analysis information. It varies the efficiency of S with the size of test cases that a method stores, as well as with the computational cost of that method [10].
- **Generality:** This parameter measures the ability of a selective re-test strategy to function in a wide and practical range of situations [10].
- **Accountability:** This parameter measures the extent to which a selective re-test strategy promotes the use of structural coverage criteria, as it increases the effectiveness of testing [10].
- **Precision:** it represents the accuracy degree of selection, which measures the extent o which a selective re-test strategy S ignores test cases that are non-modification revealing [11].

$$Percision = \frac{|T'F|}{|T'|} \quad \dots (1)$$

Where T'F is the set of failed test cases from T', which is the set of selected test cases.

- **Recall:** it represents the completeness of test selection, which measures the proportion of selected failed tests in all failed tests [11].

$$Recall = \frac{|T'F|}{|T|} \dots\dots\dots (2)$$

- **F- Measure:** This parameter evaluates the integrative benefit of the precision and recall measures by the combination of the two parameters [11].

$$FMeasure = \frac{2 * Precision * Recall}{Precision + Recall} \dots (3)$$

b) *Reduction of Test Cases*

Different parameters were emerged to evaluate the reduction techniques.

- **Test Suite Size Reduction (TSSR):** It determines the percentage of the test suite reduction by using the following equation [12]:

$$TSSR = \frac{|TSorig| - |TSred|}{|TSorig|} * 100\% \dots\dots (4)$$

Where |TSorig|, |TSred| represents the sizes of the original and reduced test suite.

- **Fault Detection Capability (FDC) Loss:** it determines the percentage of the test suite fault detection capability loss by using the following equation [13],

$$FDC = \frac{|Forig| - |Fred|}{|Forig|} * 100\% \dots\dots (5)$$

- **Percentage of Test Suite Reduction:** It represents the percentage by which the test suite was reduced from the original suite [13],

$$100 * (1 - \frac{size\ reduced}{size\ original}) \dots (6)$$

- **Fault Detection Rate**: it represents the percentage by which the rate of faults is detected [12, 13],

$$100 * (\frac{Faults\ detected\ reduced}{Faults\ detetced\ original}) \dots\dots\dots (7)$$

c) *Description of test cases reduction techniques*

In data mining, test case reduction techniques are classified into requirement based, coverage based, program slicing, genetic algorithm, greedy algorithm, hybrid algorithm, clustering and fuzzy logic [14].

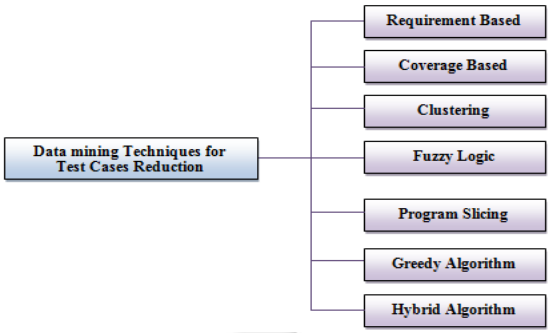

Figure 3: Data mining Techniques for Test Cases Reduction

Requirement Based

This technique used text mining in order to determine the clusters of requirements. It used k-means algorithm to cluster similar requirements after using term extraction and creating term document matrix. After the clustering was formed, the test cases clusters were formed by mapping each test case to its relevant requirements. Prioritization between test cases in the same cluster was done using the source code information, whereas prioritization between clusters was done based on the source code information along with prioritized requests from the client producing re-ordered test cases [19].

Coverage Based

Coverage based is a measure used to describe the degree to which the source code of a program is executed when a particular test suite runs. A program with high code coverage, measured as a percentage, has had more of its source code executed during testing which suggests it has a lower chance of containing undetected software bugs compared to a program with low code coverage. Many different metrics can be used to calculate code coverage.

Some of the most basic are the percentage of program subroutines and the percentage of program statements called during execution of the test suite [20].

Clustering

In order to support scalability, all test cases are clustered based on their code coverage similarity [21]. The test cases covering similar code modules are in the same cluster. Each test case is defined by a string of 0s and 1s, where each bit in the string represents a code module; 1 means that this module is covered by this test case and 0 means this module is not covered by this test case.

Distance between the defined strings of test cases is calculated using the Hamming distance [22]. Using clustering technique, close test cases (with minimum hamming distance difference) are grouped in the same cluster, indicating that they test similar modules.

Fuzzy Logic

Optimization of test suites can be achieved by using fuzzy logic. It is a safe technique and can reduce the test cases size and execution time [23]. Fuzzy logic can be used in many areas such as communication, bio informatics and experts systems. Level of testing using fuzzy logic is based on objective function quite similar to human judgment.

It can be used to make optimization in test suite for multi-objective selection criteria. It aims to find a test suite that is optimal for multi-objective regression testing and order to optimize the test suite and analyze the test suite for safe reduction which can be estimated using control flow graphs. Test cases of optimal solutions are traversed on these graphs and it is found that only fuzzy logic is safe while other approaches will be inadequate for regression testing [23].

Program Slicing

This technique is used to check a program over a specific property and to build a slice set, which is a set of statements effect to determine a statement; in many cases it is the output statement of a program, based on input values.

a) Advantages and Disadvantages of Test Cases Reduction Techniques

The advantages and disadvantages of test cases reduction techniques in data mining are given in Table I [15].

Techniques	*Advantages*	*Disadvantages*
Requirement Based	Provide a good percentage of redundancy reduction of test cases.	Some of them are time consuming and need more memory depending on how to represent the requirements
Coverage Based	Reduction rate of test cases is very high and it reduce time	For large systems the path coverage is ineffective since it consumes time and cost in identifying the coverage from a source code

Slicing techniques can help to show control-flow of a program for each test case and it is important to specify which statements are invoked with that test case. There are three types of slicing techniques,

1. Static slicing
2. Dynamic slicing
3. Relevant slicing

Using slicing techniques can decrease the number of required test cases and consequently the cost and time of testing will be decreased[13] [24].

Greedy Algorithm

The reduction process starts with the construction of test case requirement matrix which maps the test cases with the testing requirements. An association between a test case and requirement is represented by one or zero otherwise. Then the generation of the reduced test suite is made through simple mathematical operations. Greedy algorithm is used for test suite reduction also called Weighted Set Covering Technique.

It starts by determining test cases which can satisfy all the requirements. If the test case does not satisfy requirements then the algorithm repeatedly eliminate redundant test cases then update the test suite and the remaining requirements that are uncovered. The experiment is made on a test suite of Student Achievement Retrieval Navigation Model [25].

Hybrid Algorithm

Some algorithms try to reduce the number of test cases using hybrid techniques such as combine genetic algorithms and bee colony. Bee colony consists of three groups of bees: employed, onlookers and scouts. Using bees as agents the algorithm can explore the minimum set of test cases [26]. They produce a uniform representation for hybrid criteria and suggest that hybrid criteria of others can be described and that the hybrid criteria outperform the constituent individual criteria.

Clustering	produce smaller representative sets of test cases	less fault detection ability
Fuzzy Logic	A safe technique and reduce the regression testing size and execution time	Need more experiments and studies
Program Slicing	Decrease the number of required test cases and consequently the cost and time of testing will be decreased.	Need to be examined on the fault detection capability and larger generated data
Greedy Algorithm	Provide significant reduction in the number of test cases	Involve random selection of test case in a tie situation.

Hybrid Algorithm	Provide significant reduction in the number of test cases and multi-objective optimization	High complexity

Table I. Advantages and disadvantages of test cases reduction techniques

VI.CONCLUSION

Software testing is the essential and time consuming part of software development lifecycle. The time spent in testing is mainly concerned with generating the test cases and testing them. Our goal is to reduce the time spent in testing by reducing the number of test cases. For this we have incorporated data mining techniques to reduce the number of test cases. The main purpose of test case reduction is to decrease the number of test cases in order to minimize the time and cost of executing them. This paper presents the overview of the data mining approach, mainly because of its ability to extract patterns of test cases that are invisible. The paper focused on the applicability of data mining techniques in reducing the number of test cases by removing those which are redundant.

VII.REFERENCES

[1] Passant Kandil, Sherin Moussa, Nagwa Badr " A Study for Regression Testing Techniques and Tools" International Journal of Soft Computing and Software Engineering (JSCSE), Vol.5, No.4, 2015.

[2] Lilly Raamesh,G.V. Uma "Data Mining Based Optimization of Test Cases to Enhance the Reliability of the Testing " D.C. Wyld et al. (Eds.): ACITY 2011, CCIS 198, pp. 89–98, 2011. © Springer-Verlag Berlin Heidelberg 2011.

[3] B.subashini, d.jeyamala, Reduction of test cases using clustering Technique. International Journal of Innovative Research in Science, Engineering and Technology Vol 3, Special Issue 3, 2014, International Conference on Innovations in Engineering and Technology (ICIET'14). 1992-1995.

[4] Lilly Ramesh, "Knowledge Mining of Test Case System," International Journal on Computer Science and Engineering Vol.2 (1), 2009, 69-73.

[5] Mark Last and Menahem Friedman."The Data Mining approach to automated software testing."Communications of ACM, 2003.

[6] Lilly Raamesh et. al., An Efficient Reduction Method for Test Cases, International Journal of Engineering Science and Technology, Vol. 2(11), 2010, 6611-6616.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn.A Data clustering: review. ACM Computing Surveys, 31(3):264–323, 1999.

[8] Remco R. Bouckaert, "Weka Manual 3-6-1", Software manual, June 4,2009, pp-11-14.

[9] Kartheek Muthyala et al., A Novel Approach To Test Suite Reduction Using Data Mining, Indian Journal of Computer Science and Engineering (IJCSE).

[10] Bharati, C., & Verma, Analysis of Different Regression Testing Approaches. International Journal of Advanced Research in Computer and Communication Engineering,vol. 2 no.5, 2150–2155,2013.

[11] Chen, S., Chen, Z., Zhao, Z., Xu, B., & Feng, Y, Using semi-supervised clustering to improve regression test selection techniques. Fourth IEEE International Conference on Software Testing, Verification and Validation , pp. 1–10,2011.

[12] Parsa, S., & Khalilian, A., On the Optimization Approach towards Test Suite Minimization. International Journal of Software Engineering and Its Applications,vol.4 ,no. 1, pp.15–28,2010.

[13] Dr.N.Sasirekha, A.Edwin Robert and Dr.M.Hemalatha. Program slicing techniques and its applications, International Journal of Software Engineering and Application(IJSEA) Vol.2, No.3, July 2011.

[14] Kichigin, D. Test Suite Reduction for Regression Testing of Simple Interactions between Two Software Modules, Perspectives of Systems Informatics ,pp. 107–123,2010.

[15] Isha Mangal Deepali Bajaj Priyanka Gupta, Regression Test Suite Minimization using Set Theory , International Journal of Advanced Research in Computer Science and Software Engineering 4(5), May - 2014, pp. 502-506.

[16] Marwah Alian , Dima Suleiman ,Adnan Shaout "Test Case Reduction Techniques - Survey" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 5, 2016, pg- 24.

[17] http://software.nju.edu.cn/zychen/paper/2008SAC1.pdf.

[18] http://ceur-ws.org/Vol-646/EMDT2010paper4.pdf.

[19] Haider, A.A.; Rafiq, S.; Nadeem, A. "Test suite optimization using fuzzy -logic", Emerging Technologies (ICET), International Conference on, 2012, pp. 1 – 6.

[20] Arafeen, M. J., & Do, H, Test case prioritization using requirements-based clustering. Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation, ICST , pp.312–321,2013.

[21] Ur, S., Khan, R., Lee, S., Parizi, R. M., & Elahi, M., An Analysis of the Code Coverage-based Greedy Algorithms for Test Suite Reduction, The Second International Conference on Informatics Engineering & Information Science (ICIEIS2013), pp.370–377,2-13.

[22] Berkhin, P. Survey of clustering data mining techniques.Accrue Software, Inc 2002.

[23] He, M. X., Petoukhov, S. V., & Ricci, P. E. Genetic code, Hamming distance and stochasticmatrices. Bulletin of mathematical biology,Vol 66,no.5,pp. 1405-1421, 2004.

[24] Haider, A.A.; Rafiq, S.; Nadeem, A. "Test suite optimization using fuzzy -logic", Emerging Technologies (ICET), International Conference on, 2012, pp. 1 – 6.

[25] A. Mohammadian , B. Arasteh, Using program slicing technique to reduce the cost of software testing. Journal of Artificial Intelligence in Electrical Engineering, Vol. 2, No.7, 2013, pp.24-33.

[26] L. Zhang, D. Marinov, L. Zhang and S. Khurshid, An empirical study of JUnit test-ruite Reduction, 22nd IEEE International Symposium on Software Reliability Engineering, 2011, pp.170-179.

[27] B. Suri, I. Mangal, and V. Srivastava, Regression test suite reduction using an hybrid technique based on BCO and genetic algorithm, Special Issue of International Journal of Computer Science & Informatics (IJCSI), ISSN (PRINT) : 2006, 2231–5292, Vol.- II, No-1, 2

