ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Reduction of Test Cases Using Clustering Technique

Mrs.B.Subashini [#1], Dr.D.JeyaMala [*2]

Department of Computer Applications, K.L.N. College of Engineering, Pottapalayam, Sivagangai , Tamil Nadu, india.

Department of Computer Applications Thiagarajar College of Engineering, Madurai, Tamil Nadu, india.

Abstract—Software testing is a process used to identify the correctness, completeness, and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end users. The practical methods commonly used to detect the presence of errors in a program are to test it for a set of inputs called test case. A Test Case is a set of actions executed to verify a particular feature or functionality of software application. When designing the test case, the redundant test cases are formed that are of no use, increases the testing effort and increase the cost and time of testing. In this paper, the goal is to reduce the time spent in testing by reducing the number of test cases. For this the data mining approach of clustering technique is used in software testing to reduce the test suite. Mining of test case will improve the efficiency of software testing.

Keywords— Software testing, Data Mining, Clustering, Control flow graph, Test suite reduction.

I. INTRODUCTION

Testing software is a very important and challenging activity. Nearly half of the software production development cost is spent on testing. The main objective of software testing is to eliminate as many errors as possible to ensure that the tested software meets an acceptable level of quality. The tests have to be performed within budgetary and scheduled limitations. An important activity in testing is test case design. Many programming groups are relying more and more on automated testing, it require a well-developed test suite of testing scripts in order to be truly useful. If the Test suites tend to grow in size as software evolves, then testing becomes too cost to execute entire test suites. The test suite reduction techniques significantly reduce the size of the test suites. In this paper, the application of data mining techniques with software testing is used for reducing the size of the test suite [3]. The less, the number of test cases, the time taken for executing the program should also be less. This consequently improves the effectiveness of the test process [14], [5].

II. SOFTWARE TESTING

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software. A part of Software Engineering is to do Software Testing which consists of a set of activities of verifying and validating that a software application or program meets the business and technical requirements that guided its design and development and works as expected and also identifies important errors or flaws categorized as per the severity level in the application that must be fixed.

Essence of software testing is to choose a representative value (known as test case) from the input to perform the programs under test. The actual results of the programs will be checked to verify the consistency with the expected ones. If the results are different, it should take some correction, adjustment and evaluation correspondingly.

Among the different approaches we may distinguish between specification-oriented approaches (or black-box testing), which generate the test cases from the program specification, and implementation-oriented approaches (or white-box testing), which generate the test cases from the code of the program under test. Test cases have to be generated according to the test adequacy criterion, which is considered to be a stopping rule that determines whether sufficient testing has been done and provides measurements of test quality [16].

A test suite is a collection of test cases for particular software. Redundancy of test cases will be possible in the software. Redundancy is the repetition of data, between one test case and the other. So it is obvious that the reasonable structure of test suite is one of the key points in software testing achieve by which lot of time can be saved from executing redundant or unnecessary test cases[5],[7]. This replicated data isn't visible enough to capture unless and until the sophisticated techniques like data mining is used. In this paper, the proposed methodology of data mining technique is used with software testing to remove the redundant test case so that the test suites are reduced or minimized [7].

A. Test Suite

The main objective of software testing is how to select test cases with the aim of uncovering as many defects as possible. A test suite is run on the software under test and the output is examined by the tester, by comparing actual output with the expected output. If the output is incorrect then error has been discover. So, the program must be changed and testing must start again. A test suite is a collection of test cases that are intended to be used to test a software program to show that it has some specified set of behaviours [5]. Writing effective test cases for a application is a skill that can be achieved by some experience and in-depth study of the application on which test cases are being written [7].

B. Test Suite Reduction

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

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

(2) A set of testing requirements {r1, r2, r3,. . ., rn} that must be satisfied to provide the desired testing coverage of the program.

(3) 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.[5].

III. DATA MINING

Data Mining, also popularly known as Knowledge Discovery in Databases (KDD). Data mining, an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use [1],[12],[5].

A. Data Mining Concepts

There are many methods available for mining different kinds of data, including association rule, classification and clustering [12], [13].

B. Association rule

Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. For example, one may find, from a large set of transaction data, such as association rule as if customer buys (one brand of) milk, he/ she usually buys (another brand of) bread in the same transaction. Using these association rules, we can derive the association patterns from large databases [7].

C. Classification

Data classification is the process which finds the common properties among a set of objects in a database and classifies them into different classes, according to a classification model.

D. Clustering

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions [8]. Here, Clustering is one of the techniques which we are going to use for mining the test cases.

IV. EXISTING METHOD

As test suite is a collection of test cases. It is difficult to check all test cases. It takes much more time and cost is so high. Many Heuristics methods and set theories are used for the reduction of test cases [10]. Shuji Moriaski et.al proposed that defects introduced in coding/unit testing required large correction effort [19]. Yanguang Shen et.al proposed that data mining has the very great potential in software defect test research, it reduced software flaw and it promotes software credibility [18]. In the proposed work, when data mining approach is used along with software engineering it will reduce the test cases and the time and cost is saved [11], [10].

V. PROPOSED METHOD

This is an approach to join Software engineering with Data Mining. Since last few years this collaboration approach has been started with various directions for various problems. As the initiation process data mining is one of the implementation part in software engineering, it will be more efficient as a result of being more intelligent [8]. It results directly in improving software reliability [10],[11],[3].

VI. WEKA

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well suited for developing new machine learning schemes. [5]

VII. MINING TEST CASES USING DATA MINING TECHNIQUE

The test cases are created using the white box testing method,

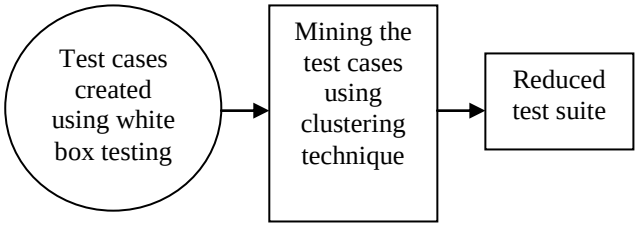


Fig. 1: Framework of Test Suite Reduction.

The basic white box testing method uses coverage criteria as a measurement of the test data. In this method, first the source code is transformed to a control flow graph. The path of the graph which is covered by test data is considered as the coverage criteria. There are three types of test data generator for coverage criteria such as path wise data generator, data specification generator and Random test data generator [16].

Let us consider a program for finding the given number is prime or not, is given below:

```
void main()
{
int number,index;
1- cout<< “Enter a number”;
2- cin>>number;
3- index=2 ;
4- while(index<=number-1)
5- {
6- if(number%index==0)
7- {
8-  cout<<“not a prime number”;
9- break;
10- }
11- index++;
12- }
13- if (index==number)
14- cout<<"prime number";
15- }
```

VIII. CONTROL FLOW GRAPH

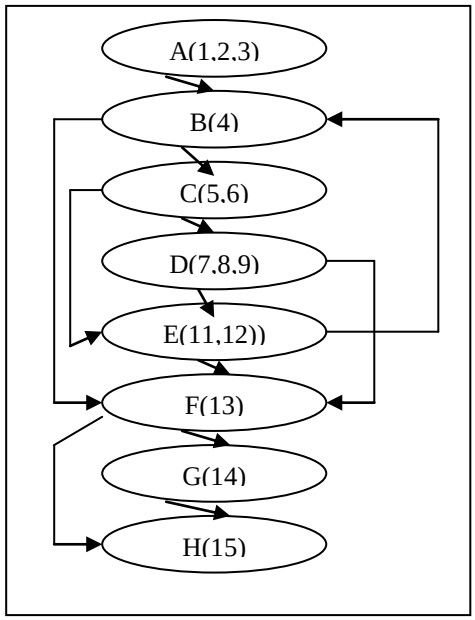
A Control Flow Graph of a program P is a directed graph G = (N, E, s, e) consisting of a set of nodes N and a set of edges E = {(n, m)|n, m ∈ N } connecting the nodes.

Each node denotes a basic block which itself is a sequence of instructions. It is important to note that in every basic block the control enters through the entry node and leaves at the end without stopping or branching except at the end. The entry and exit nodes are two special nodes denoted by s and e respectively [16].

The control flow graph for the above program is given in can be broken into DD-paths. Each DD-path is collapsed into an individual node.

The resulting graph is called a DD-path graph of the program. Every node in a DD-path graph is equivalent to a predicate. This is given in Fig 2.

Fig 2. DD path Graph for the program



The independent paths for the above program is given as

TABLE I. INDEPENDENT PATHS FOR PRIME NUMBER CHECKING

Test Case Id	Input Value	Expected Result	Independent Path
1	P1	No output is displayed	Src-A-B-F-H-Snk
2	P2	Prime Number	Src-A-B-F-G-H-Snk
3	P3	Not a Prime Number	Src-A-B-C-D-F-H-Snk
4	P4	Prime Number	Src-A-B-C-E-B-F-G-H-Snk

Likewise the independent paths for some sample programs are given and the paths are clustered using k-means clustering method. Cluster analysis groups data objects into clusters such that objects belonging to the same cluster are similar, while those belonging to different ones are dissimilar [7].

IX. K-MEANS CLUSTERING

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters

(assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids shoud be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more [12],[8].

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers[8].

TABLE II. INDEPENDENT PATHS FOR SAMPLE PROGRAMS

Program Name	**Independent Paths**	**No of Clustered Paths**
Types of triangle	Path 1: Src-A-B-D-E-F-H-J-K-M-N-O-Snk Path 2: Src-A-B-D-E-F-H-J-L-M-N-O-Snk Path 3: Src-A-B-D-E-F-H-I-N-O-Snk Path 4: Src-A-B-C-E-F-G-O-Snk	2 2 cluster paths are formed
Quadratic equation	Path 1: Src-A- B-C-H-Snk Path 2: Src-A-B-D-E-H-Snk Path 3: Src-A-B-D-F-G-H-Snk Path 4: Src-A-B-D-F-A-Snk	2 2 cluster paths are formed
Prime Number Checking	Path 1: Src-A-B-F-H-Snk Path 2: Src-A-B-F-G-H-Snk Path 3: Src-A-B-C-D-F-H-Snk Path 4: Src-A-B-C-E-B-F-G-H-Snk	2 cluster paths are formed
Biggest among Three Numbers	Path 1: Src-A-B-C-D-I-Snk Path 2: Src-A-B-C-E-I-Snk Path 3: Src-A-B-F-G-I-Snk Path 4: Src-A-B-F-H-I-Snk	2 cluster paths are formed

With the help of the clustering techniques the numbers of test cases are reduced so that the program can be checked with any one of the clustered test case and not with the entire test case that is produced by the independent paths. This will reduce the time duration in executing the test cases and it will also improve the efficiency of test case.

X. CONCLUSIONS AND FUTURE WORK

In this paper a mining approach is used to have better test cases. The better test cases can be generated, selected and are used for testing. Mining the test suite, will provide a better set of test cases. Here, the test suite reduction is done for some small programs. In the future the entire project should be taken and the test cases are reduced by making dependency between the modules.

REFERENCES

[1] Reliable Mining of Automatically Generated Test Cases from Software Requirements Specification, International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 3, January 2010.

[2] The Data Mining Approach to Automated Software Testing.

[3] SE Code Optimization using Data Mining Approach, International Journal of Computer & Organization Trends –Volume2Issue3-2012.

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

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

[6] Amalgamation of Automated Testing and Data Mining : A Novel Approach in Software Testing, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 2, September 2011

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

[8] Extracting Test Cases by Using Data Mining; Reducing the Cost of Testing, International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 3 (2011) pp. 730-737.

[9] Mining Test Cases: Optimization Possibilities, International Journal on Advances in Software, vol 5 no 3 & 4, year 2012, http://www.iariajournals.org/software/ .

[10] Automatic Software Test case Generation : An Analytical Classification Framework, International journal of Software Engineeering and its Applications, Vol 6, No.4, October 2012.

[11] Cost-Constrained Data Acquisition for intelligent Data Preparation, IEEE Transactions on Knowledge and Data Engineering, Vol 17, No 11. November 2005.

[12] Prioritizing Test Suites Using Clustering Approach in Software Testing, International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-4, September 2012.

[13] UML Generated Test Case Mining Using ISA, International Conference on Machine Learning and Computing, IPCSIT vol.3 (2011).

[14] An Efficient Algorithm for Reducing the Test Cases which is Used for Performing Regression Testing, 2nd International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2013) March 17-18, 2013.

[15] A Comparative Study of White Box, Black Box and Grey Box Testing Techniques, International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012.

[16] Different Approaches to White Box Testing Technique for Finding Errors, International Journal of Software Engineering and Its Applications Vol. 5 No. 3, July, 2011

[17] Control Flow graphs And Code Coverage, Int. J. Appl. Math. Comput. Sci., 2010, Vol. 20, No. 4, 739–749.

[18] Research on the Application of Data Mining in Software Testing and Defect Analysis", Yanguang Shen, jie Liu, IEEE Second International Conference On Intelligent Computation Technology and Automation., 2009.

[19] Defect Data Analysis Based on Extended Association Rule Mining, Shuji Moriaski, Arito Monden, Tomoko Matsumura, Fourth International Workshop On Mining Software Repositories, IEEE 2007.