Asian Journal of Applied Science and Technology (AJAST)

(Peer Reviewed Quarterly International Journal) Volume 2, Issue 4, Pages 113-123, Oct-Dec 2018

Improvisation of Fault Localization for Test Purification in Regression Testing under Database Applications

Binoop Kumar C.A[1] and A.Venugopal[2]

[1]Research Scholar, Department of Computer Science, Sree Narayana Guru College, Coimbatore - 641105.
[2]Assistant Professor, Department of Computer Science, Sree Narayana Guru College, Coimbatore - 641105.

Article Received: 25 May 2018 *Article Accepted: 27 September 2018* *Article Published: 08 October 2018*

ABSTRACT

The process of regression testing and identifying and bug fixing tends to be very costly and overwhelming time, while appending or alterations are performed on the software product. Hence diverse methods have to be projected to minimize the cost and time and the approaches include test case selection, prioritization and fault localization. The regression testing process can be made more effectual by employing the soft computing methods like machine learning and data mining where the databases are accessed by the software products. The regression testing is selected by the integration of unsupervised clustering method which consists of randomly ordered values and the associated database schema to examine the test cases that includes alterations appended to the software products related to the databases. The fault localization mechanism called "spectrum based" is executed to discover the faulty location in the source code depending on outlining the implementation of test cases. The test case purification is executed for enhancing the fault localization. The intention of the purification method is to segregate the prevailing test cases into purified test cases (smaller units) by improvising the fault localization. The two types of database applications are evaluated by the projected method with respect to various performance metrics like fault identification, test suite minimization, recall, F-measure and precision are measured. The fault localization methods like tarantula, ochiai and jaccard are proposed to analyze the effectiveness based on the above mentioned metrics are examined and the results are compared. The fault localization method turns out be very effective by performing cluster analysis for the resultant test cases.

Keywords: Fault localization, Regression testing, Test case purification, Spectrum based localization.

1. INTRODUCTION

The testing process expenses are more in regression testing that validates and identifies the altered version of the software and the newly introduced faults embedded into the former tested code can also be detected [4]. The cost of executing the entire process is very high in regression testing and hence the test cases are to be prioritized based on their imperativeness by the testing team by considering the various metrics tested earlier during the progression of regression testing. The impending target of the test case prioritization is maximizing the fault identification rate of a test suite. The fault rate identification is the process of how faster the test suite identification occurs throughout the testing. The enhanced method affords former feedback by facilitating the debugging; maximizing the likelihood and when the testing is halted earlier the highest rate of fault identification can be afforded by those test cases in the existing time. The process of regression testing plays the significant method in software development life cycle and it is used for validating the faults and to check whether the faults are embedded into the former adaptation of the software product while the latest characteristics or alterations are executed. Since the execution of the entire test cases is higher cost and hence the prioritized test cases are implemented in regression testing is the exact part of the research [17].

Execution cost of the regression testing is correlated to the verification of the product size and the implementation of the test cases which cause some restrictions while computing the existing resources for the indent. When the development of software happens to be in shorter period during repetitive or incremental versions then the developed product is said to be undesirable where the agility is misplaced [2]. Hence the agile methods can be integrated with soft computing methods to discover the test cases which make a desirable software product. In this work, fault minimization and prioritization, optimization was analyzed. Previously, fault minimization and prioritization was performed by greedy algorithms and capacity based fault identification and fuzzy entropy based optimization was examined.

Bug fixing and identifying takes more time in software development. When the bug is proposed, the testing team has to put some efforts to discover the current position of bugs in the program code and it is described in figure 1. The bug localization problem in a program is called fault localization that comprises of ranked program methods given by oracle. The fault coverage based localization also termed as spectrum-based fault localization is a family of methods that utilize the implementation outline of test cases (the coverage data) to assess the probability of faults in a program. Here, the familiar fault localization spectrum based methods like tarantula; jaccard and ochiai are used in this projected method [11]. The testing teams manually assess the program and debug the same in order to identify the bug location in the fault localization ranking method.

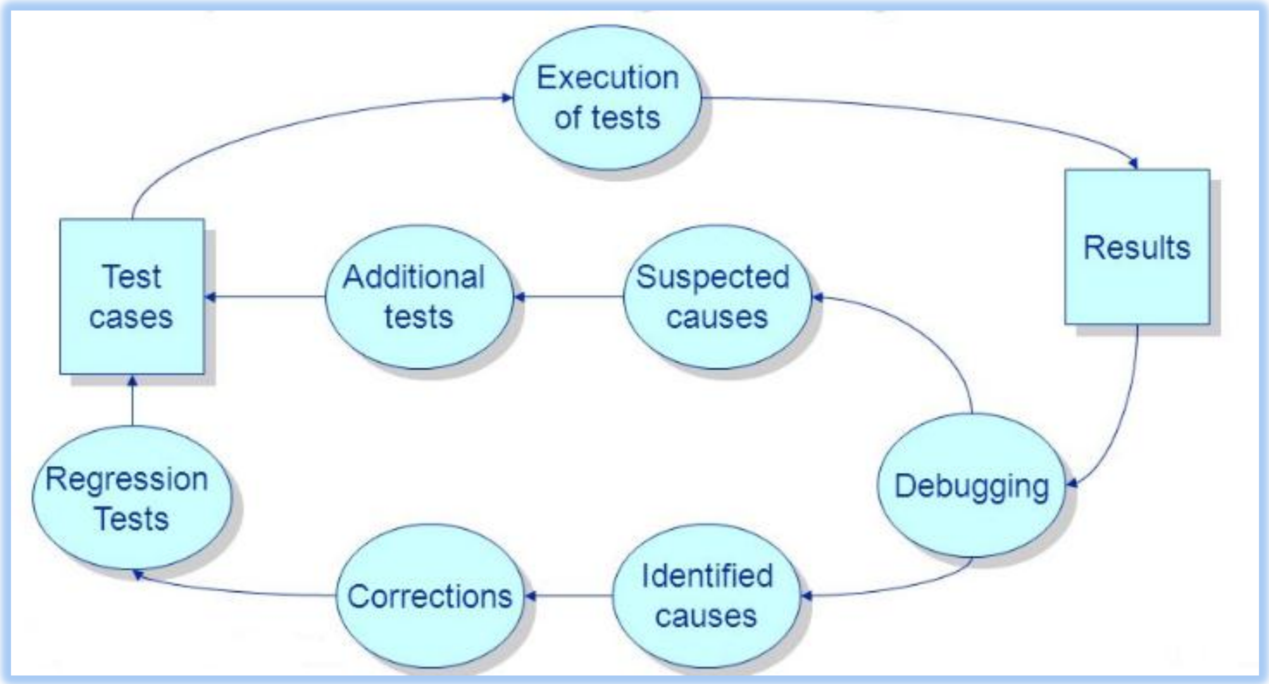


Figure No: 1 Test case evaluation in regression testing

Test cases have to be considered for fault localization. The failed test case execution can be aborted which avoids all the unaccomplished affirmations in the equivalent test case. But, the effectualness of fault localization is entirely dependent upon the number of software products. The significant suspicion is recuperating the accomplishment of avoided affirmations which gives way to more number of test cases by improving the fault localization capability. The projected method is spectrum based fault localization which is also called as test purification for enhancing the fault localization. The aim of test purification approach is create the sanitized edition of failed test cases which comprises of only one affirmation per test and the uncorrelated description of this affirmation are omitted [5]. The better fault localization can be performed on database products. Test case purification includes three parts: test case atomization and slicing and rank refinement. The test case atomization creates a group of single-affirmation test cases for every failed test case; test case slicing eradicates the disparate procedures in all the failing single-affirmation test cases; rank refinement integrates the spectrum of purified test cases with an prevailing fault localization technique and arrange the description as the final outcome (e.g., Tarantula).

The contribution of this paper includes:

1. The software products having database access in regression testing is taken and the clustering is performed for database access code. The clustering is based on the unsupervised clustering by randomly choosing a group of test cases. After clustering, the spectrum based test case purification is executed for enhancing the spectrum based fault localization. The test case manipulation is done for the better utility of prevailing test data.
2. The fault localization methods like tarantula, ochiai and jaccard are compared for the two test products estafeta and silabo with test purification.

2. RELATED WORK

Rothermel and Harrold developed the approach of software regression testing before 2 decades. Till now many diverse methods for regression testing are developed. The regression testing approaches contain the methods where it examines the source code and the test case characteristics [3]. The former methods are combined with the soft computing methods. The three methods are: Minimization: Here, P → software product and T → group of test cases connected with P. The difficulty is to attain a detachment of test cases T' that do not contain repetitive or superseded test case to validate P. Selection: the method to examine the detachment of test cases T' in a way that alterations to the software product P' are tested with T'. Prioritization: the aim is to examine the ultimate sequences of test case permutation to enhance the regression testing technique performance. Rothermel and Harrold developed a strategy to examine and identify alterations in control flow graph theory of the source and altered programs to illustrate the test cases connected to them. Chen et al. proposed a method with respect to the entity code level of various functions and the variables.

A. Pravin developed a method for the modules, this method illustrates the altered modules and their test cases, which is taken for test integration. The appliance of abstraction program into the models is applied [6]. While examining the test cases in this approach, it is noticed that some test cases have similar characteristics and connected to the similar type of faults. This method has brought more awareness among the researchers and the test cases are classified into various groups depending on prearranged measures. The works like feature measurement or metadata development are also found. Vangala et al. employed the profile level execution and static level execution where the test cases are compared by the usage of clustering algorithms. Orso et al. utilized the concept of metadata in XML format for test case selection.

Kim et al. projected a method that integrates the earlier methods of code executions with code alterations. R. Mohanty et al. developed a method that connects the outcomes of the test cases which have more number of defects and with the earlier alterations [10]. Ficco et al. developed a approach depending on the fault detection capacity level of the test cases under software iterative incremental development. The association among the set of test cases are concealed then it is very important to examine the patterns associated to these groups. Hence the optimization techniques are arises. The methods like machine learning, soft computing are utilized in regression testing. For cluster formation, the traditional data mining and machine learning methods are used. The feature of cluster analysis contains the set of test cases in regression testing. Dickinson et al. used the data mining algorithm for filtering and clustering the test cases by the program and profile execution. Some of the works are also studied regarding the cluster analysis for organizing regression testing.

With respect to data base access related to software products, A. Podgurski et al. have performed the regression testing from the perception of stored procedures using the firewall method. Willmor and Embury proposed the safe regression testing combined with the white-box and graph theory for code representation [13]. Tuya et al. projected may experiments in estimating and comparing the effectualness of various white box techniques combined with SQL descriptions with the attempt of test case selection related to the database.

V. Vangala et al. have taken the transactional feature of the database and the perception of black box testing uses the concept of classification depending upon the test case similarity [16]. Likewise, Rogstad and Briand believed that the test cases are associated to a database for aligning the database deviations where the deviations are the dissimilarities between the program version and the data manipulation and this method is similar to the entropy standard in examining the diverse clustering techniques.

The remaining sections are described as follows: section 3 describes the background and the motivation; section 4 describes the proposed work; section 5 describes the experimental results; section 6 describes the conclusion and section 7 describes the reference section.

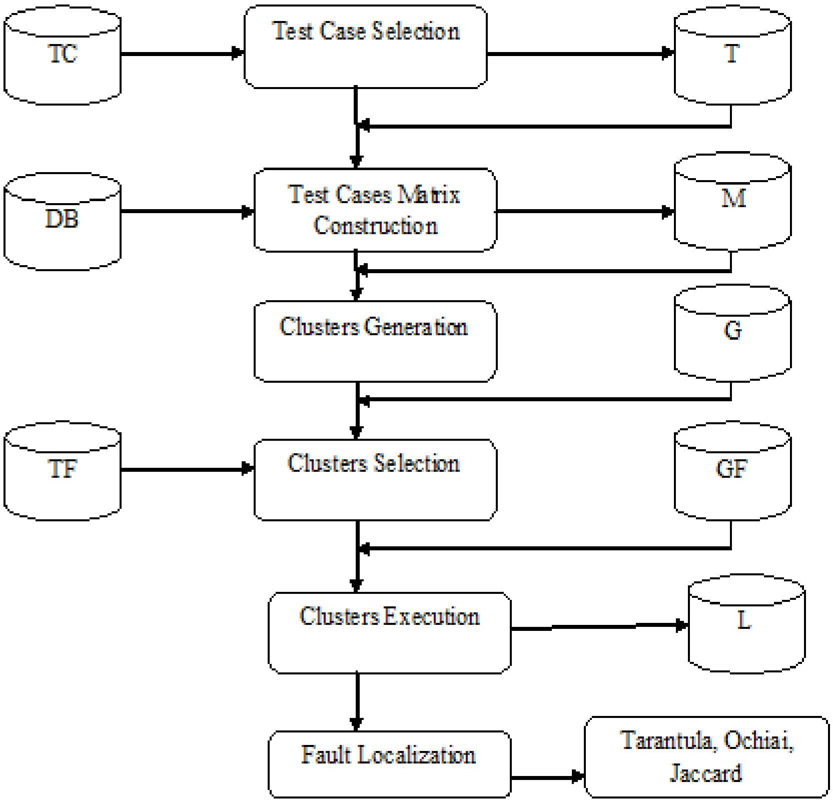
3. BACKGROUND AND MOTIVATION

The test cases are also referred to as testing methods and it is a executable part of the source code for validating the software characteristics. The test case approach consists of two methods like test input and oracle. The test input is the source data fir program implementation and test oracle examines the software accuracy regarding test input [14]. The test oracles are generated by testing team based on the expectations of business and technical fields. The test oracles are the group of executable affirmations to give the assurance regarding the software performance. A test suite is a group of test cases.

An affirmation is a binary expression that denotes the predictable program characteristics. The exception has to be thrown when the affirmation is not satisfied. When the exception happens, the affirmation is aborted and the testing structure provides the failure report. In general, a single test case is comprised of many affirmations.

A subject program is the type of program in the concept of testing and it is termed as proband or object program. The unit testing structure is the execution of the test suites automatically for the program testing [12]. The elements of the program denotes the examining the fault localization granularity. For example, the program elements can be methods, class, descriptions, etc. The paper focuses mainly on clustering of the test cases and the fault localization. The unsupervised clustering is performed for grouping the test cases and spectrum based fault localization is used for fault detection and test case purification. The test case is provided with the group of program element denotes the execution of the specific program elements. The summary of the projected are as follows: regression testing methods that are connected to database applications are to be considered. The regression testing contains the unsupervised clustering methods and it is also developed incrementally. In regression testing, the test cases are selected and construction of test case matrix is performed [15]. Based on the matrix, the clusters are generated and the clusters are selected based on the faultiness of the test cases. Then spectrum based fault localization methods are evolved in identifying the faults. The spectrum based fault localization approaches like tarantula, ochiai and jaccard are compared and the effectiveness is also compared.

4. PROPOSED WORK



TC→ test cases T→ set of test cases with access to the database DB→ database schema

M→ matrix of similarities M→ matrix of similarities G→ clusters of test cases

TF→ set of test cases with faults GF→ clusters of test cases with faults

L→ list of faults of regression testing

Figure No: 2 Regression Testing and Spectrum based fault localization

The proposed work gives the brief description related to the database applications and the fault localization algorithms used. The projected method contains test case grouping connected to the database application features performing data manipulation executed on the tables having database fields with major increments and implementing the test case groups which comprises of test cases containing faults detection. The approach is shown in figure 2.

4.1 Steps in Proposed Work

The steps performed in the proposed method are as follows:

- Selection of test cases: Initially, the test case code has to be examined that are connected to the recent feature or the alterations including the associated data bases. The textual analysis is executed for the software product code. The outcomes are a test case that has been appended to the group of test cases (T) that has the right entry to the database.
- Construction of test case similarity: In this method, the information based on the tables and the attribute values are acquired for every test case (T). The features like triggers, functions, stored procedures and views are considered. By having this information, the binary matrix (M) can be built. By constructing this matrix, the value 1 denotes the database accessing function mentioned in the test case and it gains an M_{mxn} matrix where the rows are connected to the test cases and the columns are connected to the database fields of the respective tables to determine the software product.
- Creation of test case clusters: An unsupervised clustering algorithm is utilized in this step which contains the Expectation Maximization (EM) method. It permits the statistical description analysis of the test cases called mean and variance. This information helps to examine the probability distribution function to allocate the test case membership to a cluster depending on the matrix similarity. The method will come to a decision that how much numbers of clusters are to be generated with the help of data cross-validation. The approach employs finite Gaussian mixture model where all the field values are independent random variables. After the execution of these steps, one or more clusters can be generated.
- Selection of clusters: The clusters are selected for which the clusters are to be executed while performing regression testing. The clusters are to be determined in which the current or altered test cases (GF) are situated and also the test cases are taken which reveals the faults (TF).
- Execution of clusters: The test cases (GF) are implemented where the test case related faults are created if required.
- Fault Localization: It is the process of fault management. After the execution of the faults, the accurate source of test case failure can be noticed from the group of failure indications. It is also called as test case purification method [8].

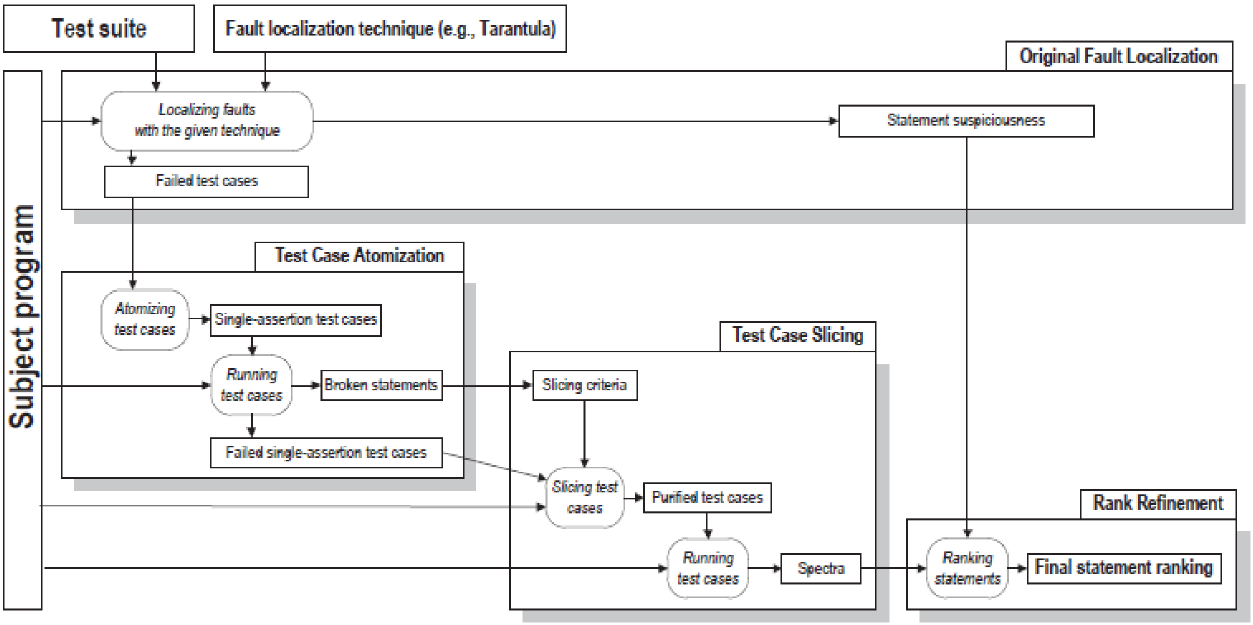


Figure No: 3 Structure for test case for purification for fault localization

The test case purification method is to create the test case refinement from every test case failure. A test case refinement is the smallest test case which contains only one affirmation and it is created by eradicating the various descriptions from the novel test case failure. Such types of test cases are purified to enhance the already prevailing methods based on fault localization. Figure 3 demonstrates the structure for purified test cases for fault localization. This organization includes three methods: atomization of the test cases, test case slicing and rank refinement [9]. The particular method based on the fault localization where the purification of test case input is the subject program along with the test suite and statement ranking is the final outcome. The provided input and the obtained output are similar as in fault localization methods. e.g., Tarantula, Ochiai and jaccard are used to test their effectiveness. The test case atomization can be performed by replacing the originally failed test cases having k number of affirmations with the k number of single affirmation test cases [1]. This type single affirmation is the replica of the novel test case by keeping the only one affirmation among the total one. The test case slicing is the method where the individual affirmation test cases are acted as a program. The dynamic slicing method is also used to eradicate the unconnected descriptions in every single affirmation test case. Finally, the rank refinement is performed by re-ranking the statements in the already prevailing fault localization methods depending on the spectrum of all test case purification [7].

5. EXPERIMENTAL RESULTS

The results can be evaluated by considering the two types of database applications. The first software product is called “Estafeta" utilized to organize the weekly schedule of the professors of an Ecuadorian university. The second software product is called “Silabo" that is employed to organize the course materials trained by the professors at Ecuadorian university.

In this work, the proposed approach is evaluated by the empirical extensive evaluation. For this method, two above two software products are required to implement in a production management.

- **Indicators and Metrics**

The effectiveness of the projected method is analyzed in order to minimize the test suites and the capability of identifying the faults are more in regression testing. The measures used in the validation are:

- **Percentage reduction of the suite (TR)**: Let T → total number of test cases with DB access. T' → number of test cases used for a regression and finally TR can be obtained by the following equation:

  $$TR = ((T - T') / T) *100 \text{ -------- } (1)$$

- **Precision(P):** Here, the value of T'f → group of all selected test cases with DB access to expose the faults, and the value of P is found by the following equation:

  $$\text{Precision} = T'f / T' \quad \text{---------- } (2)$$

- **Recall (R):** The T'f → group of test cases exposing to faults and R value is found by the following equation:

  $$\text{Recall} = |T'f| / |Tf| \quad \text{---------- } (3)$$

- **Fault detection capability:** This capability is directly associated to recall. Particularly, equation (3) is employed to compute the fault detection capability of the test case selection.

- **F-measure (F):** This approach is the integration of two measures called Precision (P) and recall (R) that is used widely in information science and the F-measure estimates the benefits and the equation 4 given by:

  $$F – \text{Measure} = (2*\text{Precision}*\text{Recall}) / (\text{Precision} + \text{Recall}) \text{ --- } (4)$$

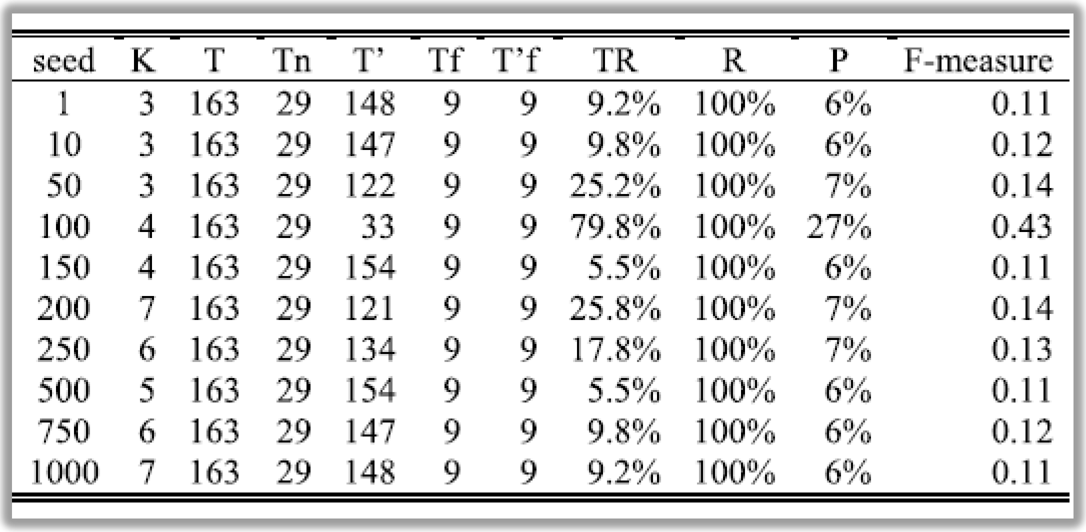
The concept of clustering is employed to assemble the test cases depending on the field similarity of the tables associated to the database. The first product (Estafeta) contains five versions and developed incrementally which contains their own test cases. The second product (Silabo) contains nine versions and developed incrementally and the number of test cases and faults are described in the table 1.

Software Product	Version	Test Cases (Tn)	Test cases for the regression test (T)	Faults (Tf)
Estafeta	1	29	-	-
	2	43	72	8
	3	62	134	11
	4	29	163	9
	5	36	199	10
Silabo	1	27	-	-
	2	33	60	8
	3	29	89	10
	4	30	119	15

	5	39	158	5
	6	49	207	7
	7	35	239	10
	8	22	267	5
	9	39	300	4

Table No: 1 Number of test cases and the fault rate

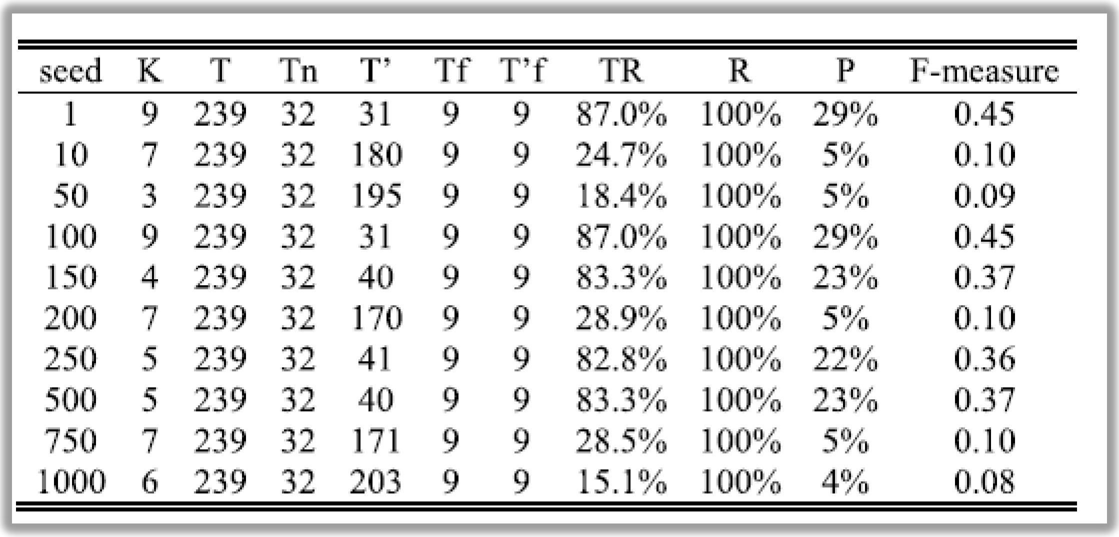
The clustering and the f-measure values of the Estafeta are described in the table 2.



seed	K	T	Tn	T’	Tf	T’f	TR	R	P	F-measure
1	3	163	29	148	9	9	9.2%	100%	6%	0.11
10	3	163	29	147	9	9	9.8%	100%	6%	0.12
50	3	163	29	122	9	9	25.2%	100%	7%	0.14
100	4	163	29	33	9	9	79.8%	100%	27%	0.43
150	4	163	29	154	9	9	5.5%	100%	6%	0.11
200	7	163	29	121	9	9	25.8%	100%	7%	0.14
250	6	163	29	134	9	9	17.8%	100%	7%	0.13
500	5	163	29	154	9	9	5.5%	100%	6%	0.11
750	6	163	29	147	9	9	9.8%	100%	6%	0.12
1000	7	163	29	148	9	9	9.2%	100%	6%	0.11

Table No: 2 clustering and F-measures of Estafeta

The clustering and the f-measure values of the Silabo are described in the table 3



seed	K	T	Tn	T’	Tf	T’f	TR	R	P	F-measure
1	9	239	32	31	9	9	87.0%	100%	29%	0.45
10	7	239	32	180	9	9	24.7%	100%	5%	0.10
50	3	239	32	195	9	9	18.4%	100%	5%	0.09
100	9	239	32	31	9	9	87.0%	100%	29%	0.45
150	4	239	32	40	9	9	83.3%	100%	23%	0.37
200	7	239	32	170	9	9	28.9%	100%	5%	0.10
250	5	239	32	41	9	9	82.8%	100%	22%	0.36
500	5	239	32	40	9	9	83.3%	100%	23%	0.37
750	7	239	32	171	9	9	28.5%	100%	5%	0.10
1000	6	239	32	203	9	9	15.1%	100%	4%	0.08

Table No: 3 clustering and F-measures of Silabo

After performing the clustering, the faults are analyzed and the fault location is determined with the help of spectrum based fault localization method. The spectrum based methods like Tarantula, ochiai and jaccard are executed for the above mentioned software products and the results are shown below in table 4.

Software Products	**Fault Localization Methods**					
	Tarantula		**Ochiai**		**Jaccard**	
	Faults	%	Faults	%	Faults	%
Estafeta	35	43.28	25	30	15	12
Silabo	39	51	21	28	15	12

Table No: 4 Fault Identification Ratios

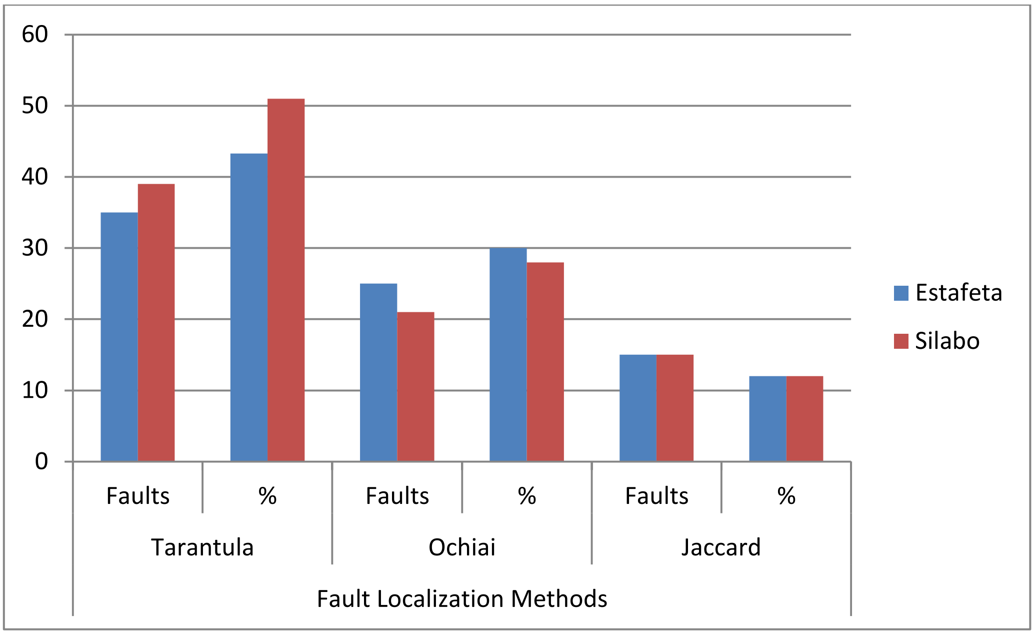


Figure No: 4 Fault Localization Ratios

The test cases are clustered to identify the faults and faults location or the test case purification can be performed by spectrum based fault localization methods. Moreover, the fault localization methods like tarantula, ochiai and jaccard are compared and the results shows that the tarantula is very effective when compared to others and it is depicted in figure 4.

6. CONCLUSION

The type of test case purification is performed for enhancing the fault localization. This work directly manipulates test cases to make better use of existing test oracles. The small fractions of test cases are generated and it is called as purified test cases, to assemble the discriminating spectrum for all affirmations in the test suite under consideration. The experimental results show that test case purification can effectually enhance original fault localization techniques. The results depicts that the advantages of test case purification exist on three fault localization techniques.

REFERENCES

[1] S. Yoo and M. Harman, ``Regression testing minimization, selection and prioritization: A survey," Softw. Test. Verification Reliab., vol. 22, no. 2,pp. 67120, Mar. 2012.

[2] R. H. Rosero, O. S. Gómez, and G. Rodríguez, ``15 years of software regression testing techniquesA survey," Int. J. Softw. Eng. Knowl. Eng., vol. 26, no. 5, pp. 675689, Jun. 2016.

[3] G. Rothermel and M. J. Harrold, ``A safe, efficient algorithm for regression test selection," in Proc. Conf. Softw. Maintenance, 1993, pp. 358367.

[4] S. Parsa and A. Khalilian, ``A bi-objective model inspired greedy algorithm for test suite minimization," in Future Generation Information Technology, Y. Lee, T. Kim, W. Fang, and D. ‘ lezak, Eds. Berlin, Germany: Springer, 2009, pp. 208215.

[5] C. R. Panigrahi and R. Mall, ``A hybrid regression test selection technique for object-oriented programs," Int. J. Softw. Eng. Appl., vol. 6, no. 4, pp. 1734, Oct. 2012.

[6] A. Pravin and S. Srinivasan, ``Effective test case selection and prioritization in regression testing," J. Comput. Sci., vol. 9, no. 5, pp. 654659, May 2013.

[7] M. Kumar, A. Sharma, and R. Kumar, ``Fuzzy entropy-based framework for multi-faceted test case classification and selection: An empirical study," IET Softw., vol. 8, no. 3, pp. 103112, Jun. 2014.

[8] H. K. N. Leung and L. White, ``A cost model to compare regression test strategies," in Proc. Conf. Softw. Maintenance, 1991, pp. 201208.

[9] D. Di Nardo, N. Alshahwan, L. Briand, and Y. Labiche, ``Coverage-based regression test case selection, minimization and prioritization:Acase study on an industrial system," Softw. Test. Verification Reliab., vol. 25, no. 4, pp. 371396, Jun. 2015.

[10] R. Mohanty, V. Ravi, and M. R. Patra, ``The application of intelligent and soft-computing techniques to software engineering problems: A review,' Int. J. Inf. Decis. Sci., vol. 2, no. 3, pp. 233272, 2010.

[11] R. O. Rogers, ``Scaling continuous integration," in Extreme Programming and Agile Processes in Software Engineering, J. Eckstein and H. Baumeister, Eds. Springer, 2004, pp. 6876.

[12] S. Biswas, R. Mall, M. Satpathy, and S. Sukumaran, ``A model-based regression test selection approach for embedded applications," SIGSOFT SoftwEng Notes, vol. 34, no. 4, pp. 19, Jul. 2009.

[13] A. Podgurski and C. Yang, ``Partition testing, stratified sampling, and cluster analysis," in Proc. 1st ACM SIGSOFT Symp. Found. Softw. Eng., New York, NY, USA, 1993, pp. 169181.

[14] S. Chen, Z. Chen, Z. Zhao, B. Xu, and Y. Feng, ``Using semi-supervised clustering to improve regression test selection techniques," in Proc. IEEE 4th Int. Conf. Softw. Test., Verification Validation (ICST), Mar. 2011, pp. 110.

[15] N. Gökce, F. Belli, M. Eminli, and B. T. Dinçer, ``Model-based test case prioritization using cluster analysis: A soft-computing approach," Turkey J. Elect. Eng. Comput. Sci., vol. 23, no. 3, p. 623, 2015.

[16] V. Vangala, J. Czerwonka, and P. Talluri, ``Test case comparison and clustering using program profiles and static execution," in Proc. 7th Joint Meeting Eur. Softw. Eng. Conf. ACM SIGSOFT Symp. Found. Softw. Eng., New York, NY, USA, 2009, pp. 293294.

[17] C. Zhang, Z. Chen, Z. Zhao, S. Yan, J. Zhang, and B. Xu, ``An improved regression test selection technique by clustering execution profiles," in Proc. 10th Int. Conf. Quality Softw. (QSIC), 2010, pp. 171179.