International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Enhanced Performance of Adaptive Random Partitioning Testing by Unifying the ARPT-1 and ARPT-2 Strategies

K. Devika Rani Dhivya, V. S. Meenakshi

***Abstract**: The software testing is considered as the most powerful and important phase. Effective testing process will leads to more accurate and reliable results and high quality software products. Random testing (RT) is a major software testing strategy and their effortlessness makes them conceivable as the most efficient testing strategies concerning the time required for experiment determination, its significant drawback of RT is defect detection efficacy. This draw back has been beat by Adaptive Testing (AT), however AT is enclosed of computational complexity. One most important method for improving RT is Adaptive random testing (ART). Another class of testing strategies is partition testing is one of the standard software program checking out strategies, which involves dividing the enter domain up into a set number of disjoint partitions, and selecting take a look at cases from inside every partition The hybrid approach is a combination of AT and RPT that is already existing called as ARPT strategy. In ARPT the random partitioning is improved by introducing different clustering algorithms solves the parameter space of problem between the target method and objective function of the test data. In this way random partitioning is improved to reduce the time conception and complexity in ARPT testing strategies. The parameters of enhanced ARPT testing approaches are optimized by utilizing different optimization algorithms. The computational complexity of Optimized Improved ARPT (OIARPT) testing strategies is reduced by selecting the best test cases using Support Vector Machine (SVM). In this paper the testing strategies of Optimized Improved ARPT with SVM are unified and named as Unified ARPT (UARPT) which enhances the testing performance and reduces the time complexity to test software.*

***Keywords**: Adaptive Testing, Random Partitioning Testing, Adaptive and Random Partitioning Testing.*

I. INTRODUCTION

Software testing [10] has been widely recognized as a standard software quality access and enhancement technique. The random generation of test cases from the set of all possible program inputs is a basic approach to testing. The main aim of the software testing is to evaluate the quality of an application and to detect software failures. In order to develop high quality software, it is essential to use software testing methods.

Revised Manuscript Received on February 28, 2020.
* Correspondence Author
K. Devika Rani Dhivya,*, Research Scholar/Assistant Professor, Bharathiar University/ Sri Ramakrishna Atrs & Science College for Women, Coimbatore, Tamilnadu, India, devika58@gmail.com.
Dr. V. S. Meenakshi, Assistant Professor, Department of Computer Science, Chikkanna Government Arts College, Tirupur, Tamilnadu, India, meenasri70@yahoo.com.

There are various software testing strategies [3][2] that available to test a software. Some of them are Random Testing, Partition Testing (PT), Random Partitioning Testing (RPT) and Adaptive Testing. RT is one of the software testing methods wherever the programs are tested by producing random, autonomous inputs. RT is broadly used in the area of software testing due to its benefits such as simplicity and easy implementation [15]. Partition Testing (PT) is one of the testing mechanisms which divides the input domains into various sum of sub-domains [4]. RPT technique is the grouping of Random and Partition Testing techniques.

The AT technique uses the history of testing to improve the effectiveness of RPT testing. AT provides the effective test selection criterion with reduced test cases, the complexity and cost of its application is higher than random and partition testing [13]. However, these testing techniques have issues such as poor effectiveness for defect detection, high time Complexity and computational cost for test case selection. These issues were resolved by using a hybrid approach called as Adaptive and Random Partition Testing (ARPT). It consists of two strategies namely ARPT-1 and ARPT-2. ARPT was already proposed to test software which utilized AT and RT in an alternative manner [11].

However, the random partitioning in ARPT has high computation complexity problem and parameters of ARPT strategies needs to be estimated for different software with different number of test cases and programs. It consumes more time and it leads to overhead process to estimate parameters of ARPT.

Hence in this research, the clustering algorithms such as Expectation Maximization (EM), K-means, Non Negative Matrix Factorization (NMF) and Self-Organizing Map (SOM) are effectively utilized in random partitioning which reduce the computational complexity of ARPT which is called Improved Adaptive Random Partition Testing (IARPT) [6]. Then, the parameters of ARPT are optimized by using Bacterial Foraging Algorithm (BFA), Artificial Bee Colony (ABC), Firefly and Improvised BAT (IBAT) optimization algorithms which reduce the number of test cases then that is said to be Optimized Improved Adaptive Random Partition Testing (OIARPT) [7]. After this, the best test cases are selected by using Support Vector Machine (SVM) machine learning algorithm called OIARPT-SVM [8]. Finally, the two testing strategies are combined as Unified ARPT (UARPT) to improve the ARPT defect detection efficiency.

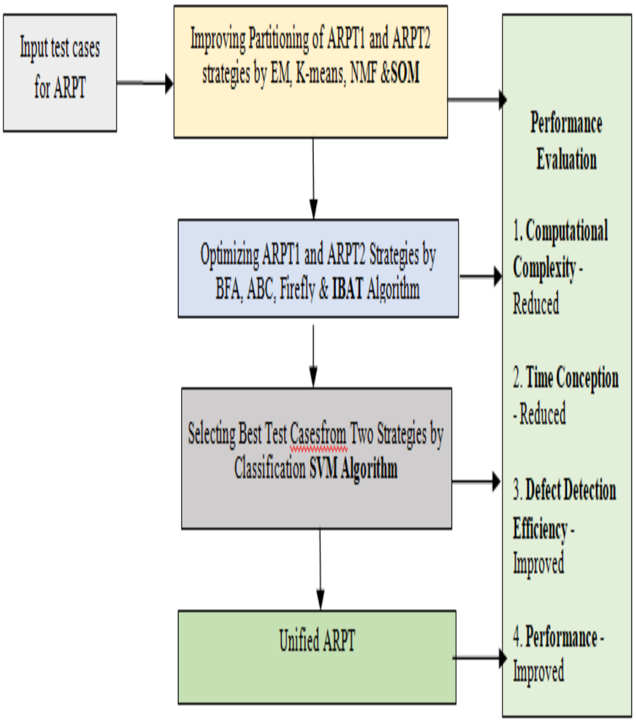

Fig 1. The Architecture Diagram

The architecture diagram of this research work is given in the following FigOIARPT-1-SVM and OIARPT-2-SVM individually select the best test cases using SVM. Then, the chosen test cases are used for identification of code defects in OIARPT-1 and OIARPT-2. OIARPT-1-SVM testing strategy alternatively uses the AT and RPT for the selected best test cases. OIARPT-2-SVM testing strategy uses AT for a specific number of best test cases and RPT is utilized for outstanding test cases. So as to improve the defect detection efficacy and to reduce the time consumption for software testing, Unified ARPT (UARPT) testing strategy is introduced in this phase of the research work. In UARPT, OIARPT-1-SVM and OIARPT-2-SVM testing strategies are combined together as a single testing strategy. For a specific number of test cases, adaptive testing is passed out and for the rest of the test cases random partition testing is done. After the accomplishment of adaptive and random partition testing, defects are detached if the failure is spotted. Then again, if the failure is not spotted then the adaptive and random partition testing completed in an elective way to detect the defects. Thus, the OIARPT-1-SVM and OIARPT-2-SVM is unified as a single testing strategy and also it improves the defect detection efficiency. The experiments are carried out in the univocityparser, marc4j and jsoup software to to demonstrate the effectiveness of UARPT regarding time utilization, defect detection efficiency, code coverage, and branch coverage.

II. LITERATURE SURVEY

Adaptive Random Testing (ART) based algorithms [14] was proposed to improve the software testing process by creating combinatorial test suites. The proposed algorithms developed test sets based on adjustable random tests and limited random testing In addition to that, the In combinatorial test spaces, the efficiency of adaptive random testing was investigated by evaluating the corresponding test suits in terms of their fault detection and size. It detects the more failures of software at the earlier stage. But the time consumption of this process is high.

A hybrid adaptive random testing strategy [12] was proposed to guarantee product consistency and trust dependencies. The Hybrid Adaptive Random Testing used some factors of test supervision and parameters of choice for test cases from different regions of size. Whereas the partition logic was also given to subdivide the areas of identical and random test inputs perfectly. An adaptive feedback was provided with some ding factors of failure rate and coverage to improve the next inputs. Based on the security levels the detection of bugs was categorized.

An approach [1] was developed to Generate path-based testing test data. The challenge faced by path coverage based testing was to build a test suite effectively which minimize the number of rejects. This issue was addressed with a novel divide-and-conquer technique grounded on adaptive random checking out strategy. The confines of an executable route were given as input to the adaptive random trying out strategy and before a tight over-approximation of associated executable path sub domain was calculated by utilizing a dynamic domain partitioning approach. The established method performed Partitioal Random Testing (PRT) based on Adaptive Random Testing (ART). A set of variables along with their variation domain were given as input to the developed approach. The resolution of the grid was need to be decided first for the partitioning the input domain. Then the developed approach was applied to generate test data.

A new Adaptive Random Testing based on Two-Point Partitioning (ART-TPP) algorithm [5] was proposed to improve the software testing process. It mainly concerned with the partitioning-based adaptive random testing. Generally, the max-area area was divided by a single point whereas in the proposed ART-TPP the max-area area was divided by the mid-point of two points. The first point of the two points was randomly generated and the second point was picked out from the candidate set based on the farthest distance criterion. However for the strip failure pattern, the proposed ART-TPP algorithm has no obvious difference with the conventional algorithms.

From above survey explores and prospects various RT, AT and ART techniques that are effective in terms of the time required specifically for the significant components, the time taken for selection testing, detection of defects and coverage of code. Implementation of any form of ART family technique involves consideration of some of the above parameters. The techniques to be related to the purely random method. This leads to the development of techniques when applying ART and this is made possible by integrating two or more techniques.

III. PROPOSED METHODOLOGY

OIARPT-SVM testing strategy depends on a basic idea where adaptive research and random partition testing techniques are used to maximize the power of each strategy. It is expected that the utilization of AT in the testing procedure would improve the effectiveness of identification of defects. The testing process can be optimized by cautiously setting the exchanging point of the testing strategies and achieving a balance between testing efficiency and time efficiency.

OIARPT-SVM consists of two variants which are defined as follows:

OIARPT-1-SVM: Using AT to pick a specific number of test cases and t afterward use RPT before changing to AT to choose a number of test cases.

OIARPT-2-SVM: Using AT to pick a certain number of test cases and fterward use RPT before changing to AT to choosenumber of test cases.

In UARPT, the two variants of OIARPT-SVM are combined together and make it as a single testing strategy. Initially the test cases are clustered based on SOM. SOM algorithm clusters the test cases based on the winning unit determined using the Euclidean distance method. Depending on the choice of the winning unit, the weights for this particular competitive learning law of the winning unit are revised. Then, the parameters of IARPT-1 and IARPT-2 testing strategy are optimized by IBAT. The IBAT optimize the parameters based on the echolocation characteristics of microbats. The IBAT generates new solution by bats flying between calculated weighted average solution and selected best solution. After maximum number of iterations, parameters of IARPT-1 and IARPT-2 are optimized.

From the clustered test cases, SVM picks the best test cases. SVM is based on the decision plane principles that describe the boundaries of the decision. A decision plane is the one that isolates between a lots of articles having distinctive class participation. The SVM demonstrating calculation finds an ideal hyperplane with the maximal edge to isolate two classes (best test cases and non-best test cases), that needs to solve the optimization problem Moreover, SVM selects the bes3t test cases with the consideration of their branch coverage and code coverage.

The selected test cases and optimized parameters are given as input to the UARPT. From the best test cases, m test cases are processed by adaptive testing and the remaining test cases are processed by random partition testing. From the results of adaptive and random partition testing, defects are removed from the software if the failure is detected. Or else if the failure is not detected, then the test cases are processed alternatively by adaptive testing and random partition testing as per the OIARPT-1-SVM testing strategy.

Adaptive testing [9] is a feedback oriented testing presenting its strong nature. It applies software cybernetics and Controlled Markov Chains (CMC). The cybernetics research studies the interplay between software and control. The software being evaluated is presented as a controlled object based on a CMC, and the testing technique acts as the appropriate controller. They form a closed-loop feedback control system together. A combination of random testing and partition testing is random partition testing. In RPT, test cases are partitioned into d subdomains $D_1, D_2, \dots D_d$ with the ith sub domain D_i composed of t_i distinct test cases. RPT choose a test case in two steps. In the first step, a sub domain is chosen based on a uniform probability distribution, i.e., the probability that each sub domain is selected is $\frac{1}{m}$. In the second step, a test case is picked from the selected sub domain and executed according to a uniform probability distribution. Suppose that the jth sub domain is selected, then the probability that any test case in the jth sub domain is picked is $\frac{1}{t_j}$. The adaptive testing and random partition testing is used in UARPT testing strategy to test a software.

```
UARPT Approach
Input: Test cases
Output: Defect detection
Cluster test cases based on SOM technique.
Optimize OIARPT parameters using IBAT and select the best test cases using SVM.
while (stopping criterion is not satisfied)
    if (x < y)
        T_x = adaptive();
    else
        T_x = random partition();
    end if
    execute (T_x);
    if (failure == True)
        removeDefect();
    else
        x++;
        if (x < k(S))
            T_x = adaptive();
        end if
        if (k(S) < x ≤ k(S) + l(S))
            T_x = random partition();
        end if
        execute(T_x);
        if (failure == True)
            removeDefect();
            Sig_R(S) = 1;
        end if
        if (S == k(S) + l(S))
    (k(S+1), l(S+1)) = CalcSteps(k(S), l(S), Sig_R(S));
            S = S + 1; x = 0; Sig_R(S) = 0;
        end if
    end if
end while
```

IV. RESULT AND DISCUSSION

The results of the experiments are conducted to demonstrate the effectiveness of the proposed optimization algorithms in the strategies of ARPT testing.

This proposed research work is conducted in three different software named as univocityparser, marc4j [10] and jsoup.

Table 1. Result Comparison

Time Consumption (ms)			
Testing strategies	**Software**		
	Univocityparser	**marc4j**	**Jsoup**
OIARPT-1-SVM	2	1.9	1.9
OIARPT-2-SVM	2.2	2.19	2.2
UARPT	**1.6**	**1.5**	**1.6**
Defect Detection Efficiency (DDE) (%)			
Testing strategies	**Software**		
	Univocityparser	**marc4j**	**jsoup**
OIARPT-1-SVM	93	92	92
OIARPT-2-SVM	90	89	90
UARPT	**96**	**95**	**94**
Code Coverage (%)			
Testing strategies	**Software**		
	Univocityparser	**marc4j**	**jsoup**
OIARPT-1-SVM	82	83	80
OIARPT-2-SVM	77	79	78
UARPT	**95**	**96**	**94**
Branch Coverage (%) (for 30 programs)			
Testing strategies	**Software**		
	Univocityparser	**marc4j**	**jsoup**
OIARPT-1-SVM	90	88	92
OIARPT-2-SVM	88	85	90
UARPT	**93**	**91**	**93**

From Table 1, it is proved that the proposed UARPT testing strategy has better time consumption, DDE, code coverage and branch coverage than the OIARPT-SVM testing strategies.

A. Selected test cases

An efficient testing strategy detects the defects in software with minimum number of test cases and high coverage.

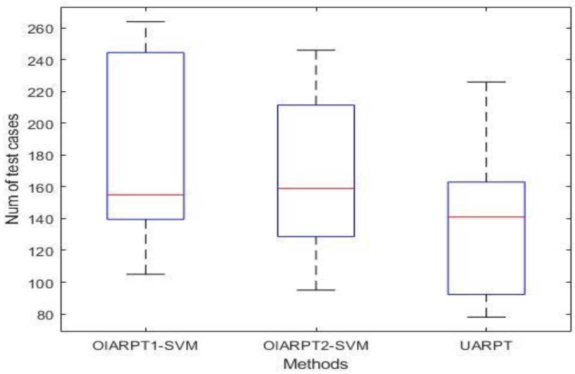

Fig 2 Comparison of Selected test cases for univocityparser software

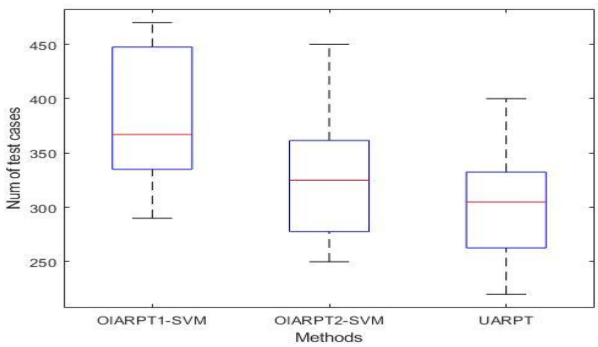

Fig 3 Comparison of Selected test cases for marc4j software

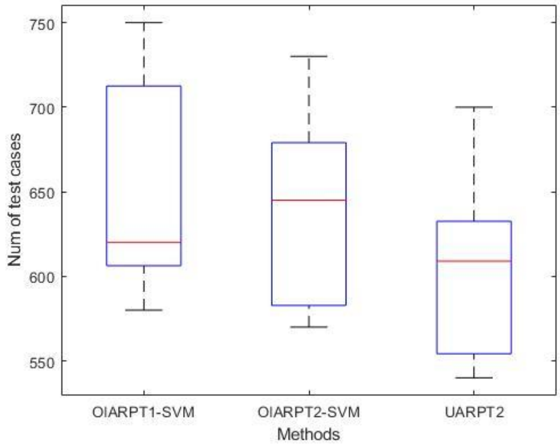

Fig 4 Comparison of Selected test cases for jsoup software

Fig 2,3 and 4 shows the number of test cases selected by OIARPT-1-SVM, OIARPT-2-SVM and UARPT testing strategies for univocityparser, marc4j and jsoup software. The red line indicates at that number of test cases the defects in the software is detected. From the results of selected test cases, it is proved that the UARPT detects the defects with less number of test cases than the other testing strategies.

V. CONCLUSION

In this paper, UARPT is proposed to combine the OIARPT-1-SVM and OIARPT-2-SVM testing strategies which enhance the testing performance. Initially, the test cases are clustered based on SOM technique and the parameters of ARPT testing strategies are optimized by using IBAT. Then, the computational complexity of testing strategies is reduced by selecting the test cases by using SVM. The selected test cases are given as input to UARPT. In UARPT, OIARPT-2-SVM is processed and then OIARPT-1-SVM process is carried out when the failure is not detected by OIARPT-2-SVM testing strategy. Thus the OIARPT-1-SVM and OIARPT-2-SVM are combined which enhance the defect detection efficiency. The experimental results show that the proposed UARPT has better time consumption, DDE,

code coverage, and number of test cases selected than other testing strategies for univocityparser, marc4j and jsoup software.

REFERENCE

1. Esmaeel Nikravan, Farid Feyzi and Saeed Parsa. “Enhancing path-oriented test data generation using adaptive random testing techniques”, 2nd IEEE International Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 510-513, 2015.
2. Babu Zachariah. “Analysis of Software Testing Strategies Through Attained Failure Size”, IEEE Transactions on Reliability, vol. 61, no. 2, 2012 pp. 569-579.
3. Changhai Nie, Huayao Wu, Xintao Niu, Fei-Ching Kuo, Hareton Leung and Charles J. Colbourn. “Combinatorial testing, random testing, and adaptive random testing for detecting interaction triggered failures”, Information and Software Technology, vol. 62, 2015, pp. 198-213.

4. Chang-Ai Sun, Hepeng Dai, Huai Liu, Tsong Yueh Chen, and Kai-Yuan Cai. “Adaptive Partition Testing”, IEEE Transactions on Computers, vol. 68, no. 2, 2018, pp. 157-169.
5. Chengying Mao. “Adaptive random testing based on two-point partitioning”, Informatica, vol. 36, no. 3, 2012.
6. Devika Rani Dhivya K., and V. S. Meenakshi, “Improved Time Performance of Adaptive Random Partition Software Testing by Applying Clustering Algorithms”, Intern. J. Adv. Res. Comput. Sci., vol. 8, 2017.
7. Devika Rani Dhivya K., Meenakshi V.S.,”An Optimized Adaptive Random Partition Software Testing by Using Bacterial Foraging Algorithm”, Computational Vision and Bio Inspired Computing. Lecture Notes in Computational Vision and Biomechanics, vol 28. DOIhttps://doi.org/10.1007/978-3-319-71767-8_47, Publisher Name Springer, Cham, Print ISBN978-3-319-71766-1, Online ISBN978-3-319-71767-8. Pg.No. 542-555, 2018
8. Devika Rani Dhivya K., and V. S. Meenakshi “Selection of best Test Cases using Support Vector Machine for ARPT “, International Journal of Recent Technology and Engineering, Volume-8 Issue-2S8, August 2019, ISSN: 2277-3878 (Online).
9. Hai Hu, Chang-Hai Jiang and Kai-Yuan Cai. “Adaptive software testing in the context of an improved controlled Markov chain model”, 32nd Annual IEEE International Computer Software and Applications Conference, 2008, pp. 853-858.
10. J Lee, S Kang and D Lee. “Survey on Software testing Practices”, IET Software, vol. 6, no. 3, 2012, pp. 275-282.
11. Junpeng Lv, Hai Hu, Kai-Yuan Cai and Tsong Yueh Chen. “Adaptive and random partition software testing”, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 12, 2014, pp. 1649-1664.
12. Mahesh Malviya and Reetika Patidar. “H-ARPT: A Hybrid Adaptive Random Partition Testing for Improved Bug Detection and Test Coverage”, International Journal of Computer Science and Information Technologies, vol. 5, no. 5, 2014, pp. 6796-6801.
13. Peng Xiao, Yongfeng Yin, Bin Liu, Bo Jiang and Yashwant K. Malaiya. “Adaptive testing based on moment estimation”, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, pp. 1-12.
14. Rubing Huang, Xiaodong Xie, TsongYueh Chen and Yansheng Lu. “Adaptive Random Test Case Generation for Combinatorial Testing”, 36thAnnual IEEE conference on Computer Software and Applications, 2012, pp. 52-61.
15. Zhibo Li, Qingbao Li, and Lei Yu, “An Enhanced Adaptive Random Testing by Dividing Dimensions Independently,” Mathematical Problems in Engineering, vol. 2019, Article ID 9381728, 15 pages, 2019.