Journal of Theoretical and Applied Information Technology
15th May 2021. Vol.99. No 9
© 2021 Little Lion Scientific

IMPROVING REGRESSION TESTING QUERY REPLYING PROCEDURE USING SECURE OPTIMIZED GRAPH WALK SCHEME

U.SIVAJI[1], Dr. P. SRINIVASA RAO[2]

[1]Research Scholar, Department of Computer Science and System Engineering, Andhra University, Vishakhapatnam, Andhra Pradesh, India.

[2]Professor, Department of Computer Science and System Engineering, Andhra University, Vishakhapatnam, Andhra Pradesh, India

E-Mail: sivajiu17@gmail.com

ABSTRACT

Day by day the creation of software and application is growing rapidly, so that testing the software's or codes for its faults are the key paradigm in digital field. However, testing the software is not at all an easy role, the huge size of data made execution difficulties. To overcome these issues, the current article aimed to develop a novel Optimized Hash Graph walk Ant Lion Replica (OHGW-ALR) to schedule the test cases based on priority manner. Moreover, while arranging the test cases in priority based it minimize the usage of resource and processing time. Moreover, the presence fitness process of Ant lion model in graph walk scheme has improved the fault prediction performance. Finally, the proficient rate of the developed replica is verified by comparing the key metrics with other associated works. In that the developed OHGW-ALR has gained better exact rate of fault predict.

Keywords: *Regression Test, Fault Detection, Flaw Mitigation, Test Case, Priority Based Scheduling*

1.INTRODUCTION

Nowadays, several digital appliances were developed for numerous specific processes. However, several issues are raised because unexpected outcome [1]. So that, inspection scheme is needed to verify each application based on its behaviours [2]. For this reason, the replica regression testing is executed in several platforms [3]. In addition, the regression testing flow is carried in six stages [4],

- Selection, minimization or priority based test case
- Test arrangement or set up
- Test case ordering [23]
- Execution of test case
- Progression assessment [25]
- Flaw mitigation

The above detailed sub events are executed successfully to achieve the regression test target [3].

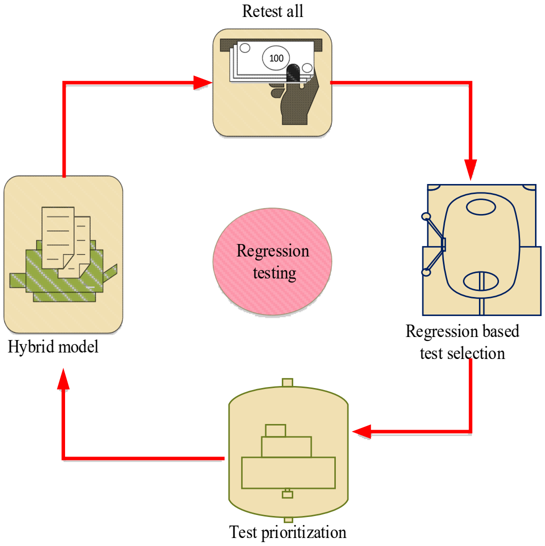

Fig.1 Frames Of Regression Testing

In addition, the developed software often required some maintenance feature to enhance its process [5]. So, testing is the most need function to avoid the new bugs while code execution or software maintaining [6]. The key factor in digital appliances is data integrity, thus the parameter data integrity is successfully processed using checksum replica models [7]. In test case analysis, the priority based test case scheduling is the key process to make the regression investigation function easier [8]. Furthermore, the purpose of this regression inspection is to predict the software or coding bugs in earlier stages to improve the system functions [9]. Henceforth, the regression based testing scheme has functioned with diverse kind [10] of testing models including box testing [11]. Also, testing of software is automatic process that processed in the software back ground [12]. In addition, there are 2 types of testing replicas [13] that are priority and vision [14]. The vision based inspection strategy includes software types, version and so on [15]. But the priority based module is functioned based on the most important case and period based cases [16].

Several security frames were discussed for regressing investigation such as dynamic call [17], horizontal model [18], and so on to improve the confidential measure in regression frame. But still preserving the high confidential rate is challenging. So that, the current research work planned to design an efficient frame model to enhance the fault prediction accuracy and to diminish the processing time.

The chief steps of the developed replica is described as follows,

- Primarily, n amount of test cases are trained to the system
- Developed a novel OHGW-ALR to advance the regression testing by ordering the test cases based on priority
- Once the test cases are collected that is saved in cloud memory
- Consequently, the save statistics are secured under the binary approach
- The present faults are predicted, here the presence of AL fitness improves the detection accuracy
- Finally, the developed replica is rated in terms of exactness rate, recall, precision, processed test case and so on, and also attained best results.

The structure of this article is designed as sector 2 details associated literatures, consequently, the system frame and issues definition is described in section.3, sector,4 detailed a novel approach, then progress of developed replica is explained in part.5 and the arguments is concluded in section.6.

2.RELATED WORK

The key reason for developing the regressing replica in programming is to recognize the execution flaws in earlier stage. Jianlei Chi *et al [17]* projected a dynamic call series to identify the software faults in earlier period. In addition, the regression scheme is more appropriate in several test cases because of iterative function. However, this dynamic replica takes more duration to finish the job.

Different software module has various programming platform, thus the code is diverged based on specific software frames. So, it causes a wide range of difficulties in regression inspection. For that, Juan Pablo Sandoval Alcocer*et al* [18] designed a Horizontal Profiling strategy to inspect the code execution and to find the flaws. Moreover, for testing benchmark dataset is adopted. However, it has preserved very less measure of exactness.

MostafaMahdieh *et al [19]* projected a flaw analysis regression based approach is utilized in software modelling to examine the own characteristics. In addition, the usage of flaw analysis has enhanced the system process by examine the flaw rate in earlier stage. Finally, the designed replica was validated with other associated works. But, fault based regression scheme is hard to design.

The regression examine scheme is flexible for all applications, so NasirMehmoodMinhas *et al* [20] projected a types of regression system in embedding frame model to point out the successive measure of the developed frame. Moreover, several queries are designed to estimate the testing system efficiency. However, the planned schema has obtained very less precision value.

To enhance the testing proficiency of regression, priority based scheme is developed. So HONGDA WANG *et al* [21] designed a priority based test frame in form of coverage replica. After the execution of this developed model, its parameter is estimated with other associated framework to update its efficiency measure. By the estimation, the designed paradigm has gained high accuracy. But during the continuous iteration it has obtained very high error measure. Hence, the statistics of associated literatures are detailed in table.1.

Table.1 Assessment Literature Survey

Author	Techniques	Advantage	limitations
Jianlei Chi *et al [17]*	dynamic call series	software faults are recognized with high exactness measure	time complexity
Alcoceret *al* [8]	Horizontal profiling model	it has the capability to process large datasets	less accuracy measure
MostafaMahdiehet *al [19]*	regression flaw analysis	it has recognized the flaw measure in an earlier stage	hard to design
NasirMehmoodMinhase*t al* [20]	regression system in embedding frame	several queries are answered successfully	less exactness rate
HONGDA WANG *et al* [21]	coverage replica	attained high accuracy	high error rate

3. SYSTEM MODEL AND PROBLEM DEFINITION

The regression based software inspection process is carried out using 4 types, minimization based, selection based, retest and priority based model [22]. Moreover, the basic function of regression testing is elaborated in fig.2.

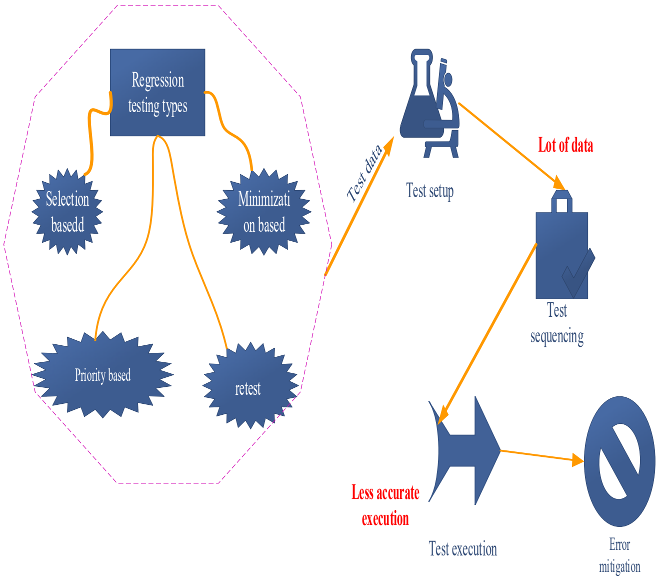

Fig.2 System Model And Problem Definition

In some case, the regression test cause bit error. At that period if the bit flaw is high then it has pertained very less execution rate and take more execution period. So that, the proposed investigation article aimed to create an efficient regression test scheme based on priority scheduling.

4. PROPOSED APPROACH: OHGW-ALR

Nowadays, in digital area verifying the software performance is the significant task to improve its processing rate. So that several testing models are implemented: however, in some cases difficulties were raised in testing environment. So, the present research work focused priority based test case system to investigate the system functions. To execute the priority based case testing in software environment, scheduling replica is more adoptable.

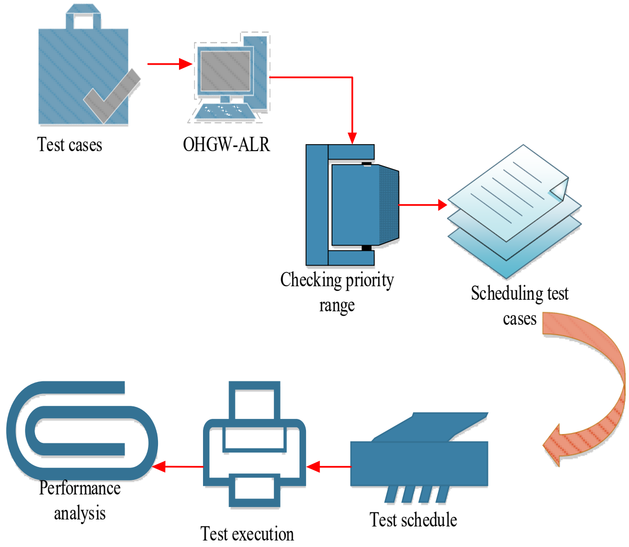

Fig.3 Proposed: OHGW-ALR

So, the current research article planned to develop Optimized hash Graph walk Ant Lion Replica (OHGW-ALR) to schedule the test cases and to mitigate the flawrate. Initially, the test cases are protected underany kind of checksum hashing model. The projected architecture is described in fig.3.

4.1 PRIORITY BASED TEST CASE ORDERING

The reason of functioning priority based ordering scheme is reduce the resource usage and traffic measure. To schedule the test cases, the Ant Lion (AL) model [26] is taken; it is the kind of optimized framework. Here, it is worn to order the test cases. In addition, here, the test cases are processed based on priority manner. The elevation node is estimate using the Ant Lion (AL) random walk fitness by eqn. (1).

$$n^{*g}_{h} = \frac{(n^{*g}_{h} - s)\times(d - C^{g}_{h})}{(d-s)} \tag{1}$$

Here, the prioritization of test frames is processed based period, which means deadlines. Moreover, a specific required duration is fixed for each test case process. During the execution function, if the required time of progress is less that the average period then it's considered as first priority test case. Thus the priority based test case choosing function is processed using eqn. (2).

$$test \quad case = avg \quad time > deadline \tag{2}$$

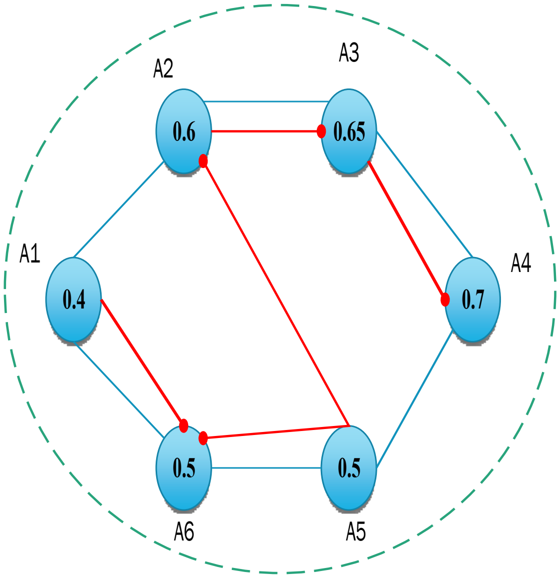

Fig.4 Six Diverse Test Cases

Here, the six hubs A1, A2, A3, A4, A5, and A6 represent 6 different test cases, that is exposed in fig.4. Moreover, the execution of each test cases based on its deadline. Let us consider, 0.5, 0.6, 0.65, 0.7 are the output required times duration. In addition, based on priority, the test case of A1 is processed first.

$$fault = \frac{f + f^{*}}{2} \tag{3}$$

Once, the flaw is predicted then its compressed using eqn.(3). Also, the working system of the projected replica is explained in algorithm.1.

Algorithm 1: pseudo code of OHGW-ALR

```
start
{
    int: $W, W^{*}$ , $n^{*}$
    // here, $W$ is the weighted graph and
    $w^{*}$ is the non-weighted
```

```
        $n^*$ = present hub in network medium

         int $i, j, k$

        // here, the test case parameter is
        determined as $i, j, k$

          $i$ =flaw mitigation parameter

          J=priority based test selection

          K= collected test cases

          // determining regression analysing
          parameter.

           $k$ = number   of   test   cases

          // all collected test cases are
          scheduled using  $j$

      checksum ()

          K= checksum

           $j$ = short period cases

          // ant lion behaviour is updated to
          select  the primary test case

          Scheduling cases= short deadline

      Test execution()

          int $t$

          // executing test case $t$

      Fault detection()

          $i$ = recorded     flaw

          Optimized flaw record using eqn.(3)

}

stop
```

Query 1: Is the developed OHALR strategy schedule all test circumstances?

Query 2: Is the projected OHALR replica mitigates the flaw in execution?

Query 3: Is the developed model executes the test cases in priority based?

4.2 SECURING THE TEST CASE BY CHECKSUM

Several binary replicas are flexible in digital field, so to secure the test operation is crucial thing in regression based inspection cases. Because, the regression based examine model can verifies the system performance using the systems behaviour and cover statistics such as memory size capacity, speed and so on. For that securing the dataset are needs in verification process. So that, once the cases of test is gathered from specific software platform then it is protected by a specific checksum replica that is binary scheme. Thus the process of hashing frame is detailed in fig.5 and fig.6.

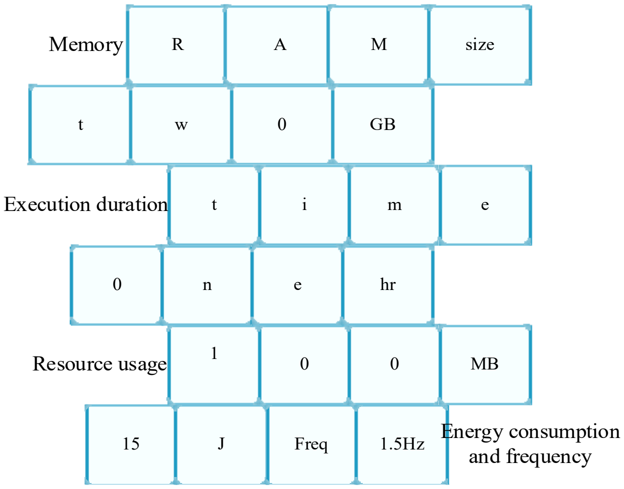

Fig.5 Gathered Test Cases

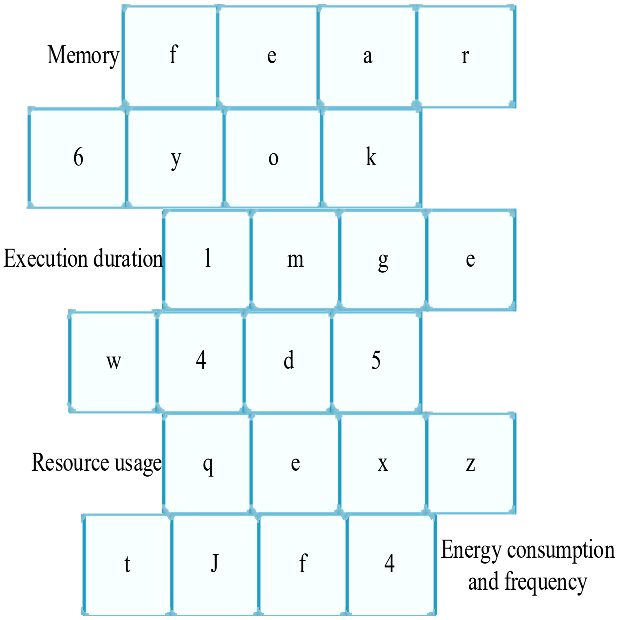

Fig.6 Collected Test Case After Hashing

Hereafter, the collected test info is secured by diverse hashing approaches. In addition, after the checksum function, the collected data is changed as code form that is exposed in fig.6. Thus, when the malicious event is located in the network medium while the regression analysis, they count not able to record the statistics. The interior function of the designed framework is revealed in fig.7

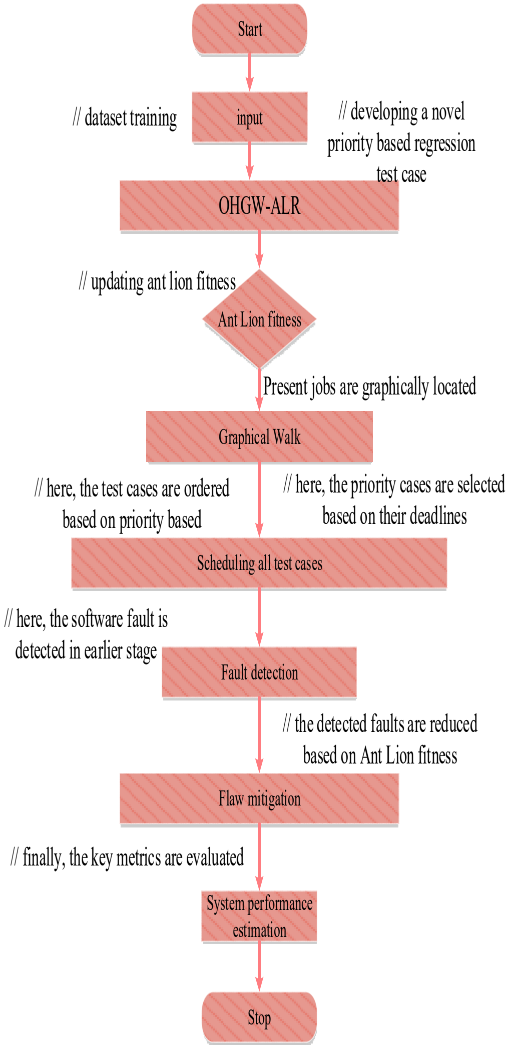
Start
// dataset training
input
// developing a novel priority based regression test case
OHGW-ALR
// updating ant lion fitness
Ant Lion fitness
Present jobs are graphically located
Graphical Walk
// here, the test cases are ordered based on priority based
// here, the priority cases are selected based on their deadlines
Scheduling all test cases
// here, the software fault is detected in earlier stage
Fault detection
// the detected faults are reduced based on Ant Lion fitness
Flaw mitigation
// finally, the key metrics are evaluated
System performance estimation
Stop

Fig.7 Flow Model Of Projected Strategy

Answer (query 1): All the present cases are scheduled based on priority manner; here the test priority is selected based on its deadline. In addition, the cases that have very short duration for execution, is considered as primary priority. Answer (query2): By the support of ant lion fitness, the regression flaw is mitigated. Answer (query 3): Yes the designed scheme is processed based on priority. Thus to analyze, the performance of regression examine scheme the query and answering framework is updated. Finally, during the query satisfaction, the proficient rate of developed test case replica is determined.

5. RESULT AND DISCUSSION

To investigate the system function, an efficient regression based test case replica is important.

Henceforth, the current research aimed to design a novel OHGW-ALR framework to inspect the system process and to detect the present flaws. In addition, the chief aim of this test case frames it to enhance the specific system by identifying its flaw in earlier stages. The successive measure of the developed approach is estimated with other relevant research works with several parameters. Thus, the major metrics are validated with other associated models like coverage based model (CBM), Additional Greedy call model (AGC) and Horizontal Profiling (HP).

5.1 Flaw detection

The purpose of performing regression based test case model is to predict the flaws in specific software frame in an earlier stage. If the flaw is detected in an earlier period then a suitable reduction scheme is designed to minimize the flaw rate.

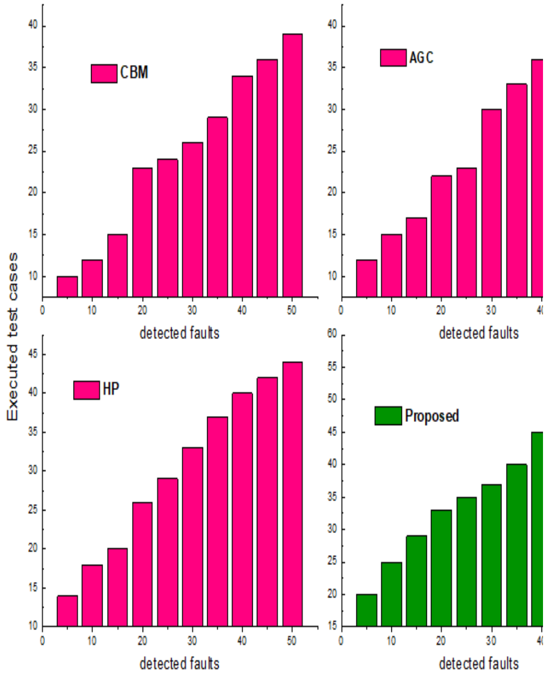

Fig.8 Detected Flaw Rate With Executed Test Cases

Here, the maximum amount of processed test cases is 55. Simultaneously, the model HP has executed 44test cases, AGC has handled 40 samples of and the approach CBM has processed 39 numbers of cases. Moreover, the attained statistics assessment is revealed in fig.8 and table.2.

Table.2 Test execution vs detection flaw

fault detection	Test case execution			
	CBM	AGC	HP	Proposed
5	10	12	14	20
10	12	15	18	25
15	15	17	20	29
20	23	22	26	33
25	24	23	29	35
30	26	30	33	37
35	29	33	37	40
40	34	36	40	45
45	36	38	42	50
50	39	40	44	55

5.2 Accuracy

The exact rate of flaw detection improved the method performance. The technique, which has gained less exact measure, has the worst performance. By the assessment statistics, the developed scheme has achieved the exactness of fault detection is 99%. This is maximum accuracy rate of the designed frame. Hence, the statistics of accuracy is detained in table.3 and fig.9. Also, the measure of exactness is estimated by eqn. (4).

$$Accuracy = \frac{exact \quad detection}{total \quad prediction} \tag{4}$$

Table.3 Accuracy assessment

	Accuracy			
fault detection	CBM	AGC	HP	Proposed
5	93.8	95	96	99
10	93.3	94.7	95.5	98.9
15	93	94.2	95	98.6
20	92.8	93.8	94.8	98.3
25	92.5	93.3	94.3	98
30	92	93	94	97.9
35	91.6	92.8	93.6	97.6
40	91	92.5	93.3	97.3
45	90.6	92	93	97
50	90.3	91.6	92.6	96.8

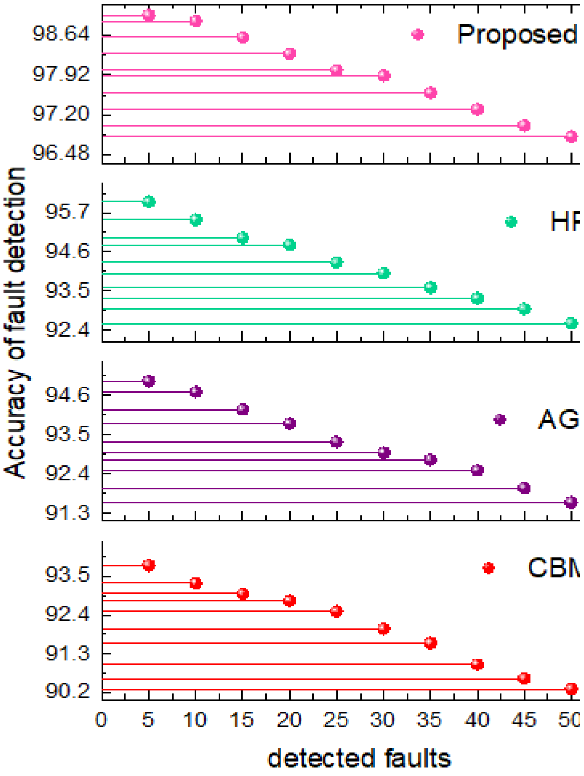

Fig.9 Comparison Of Exactness Measure

5.3 Precision

The supporting metrics of accuracy estimation is precision, the replica that has gained better exact measure has high precision rate. Furthermore, the progress of precision rate is detailed in table.4 and fig.10.Also, the measure of precision is assessed by eqn. (5).

$$precision = \frac{correct\ \ detection}{correct\ \ detection + false\ \ dete} \qquad (5)$$

Table.4 Precision assessment

	Precision			
fault detection	CBM	AGC	HP	Proposed
5	93.7	95	95.9	98.9
10	93.1	94.6	95	98.91
15	92.9	94.3	94.9	98.5
20	92.5	93.4	94.6	98.31
25	92.4	93.1	94.3	98.1
30	92.1	92.8	94.1	97.8
35	91.6	92.5	93.7	97.5
40	91.2	91.5	93.4	97.31
45	90.5	91	93.1	97.1
50	90.3	90.6	92.5	96.8

Here, the maximum precision gained by a novel approach is 98.9%. Simultaneously, the model HP has gained 95.9% of precision measure, AGC has pertained 95% of precision rate and the approach CBM has achieved 93.7% of precision measure.

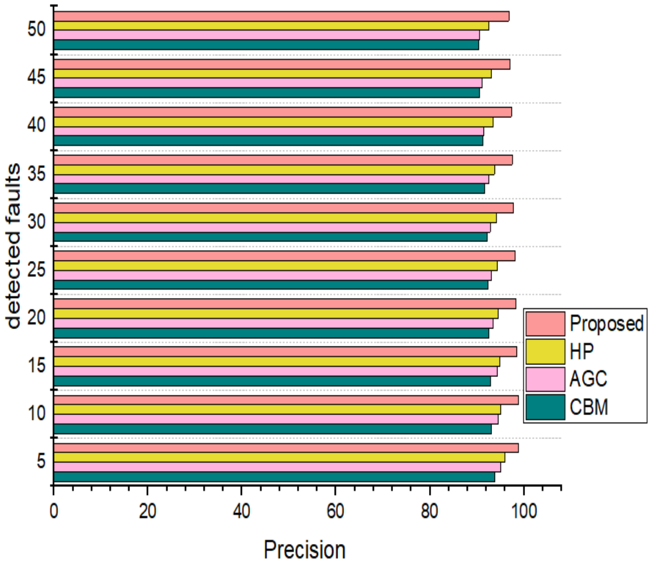

Fig.10 Assessment Of Precision

5.4 Recall

By the exact and false prediction recall is evaluated, by the evaluation the achieved recall measure is almost unique to the exactness measure. Thus, the efficiency of the developed scheme is verified.

Table.5 measure of recall

fault detection	Recall			
	CBM	AGC	HP	Proposed
5	93.6	95	95.9	98.9
10	93.1	94.6	95	98.91
15	92.6	94.32	94.91	98.51
20	92.5	93.4	94.61	98.32
25	92.4	93.2	94.3	98.2
30	92.1	92.7	94.11	97.8
35	91.5	92.5	93.7	97.51
40	91.3	91.5	93.41	97.31
45	90.5	91.1	93.1	97.1
50	90.3	90.6	92.5	96.8

Here, the greatest recall gained by a novel approach is 98.9%. Simultaneously, the model HP has gained 95.9% of recall measure, AGC has pertained 95% of recall rate and the approach CBM has achieved 93.6% of recall measure.

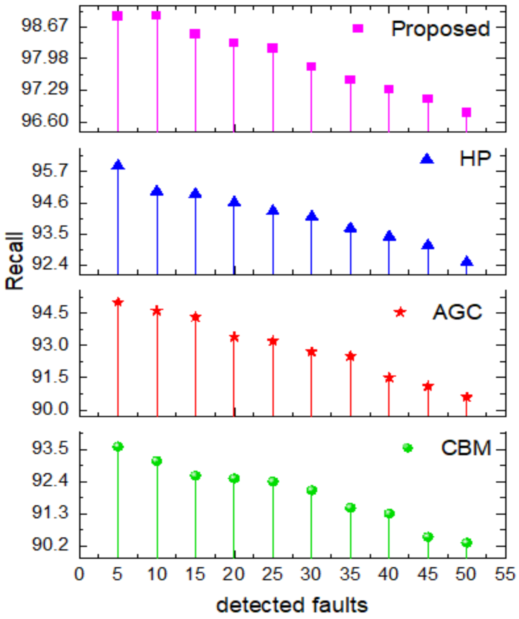

Fig.11 Assessment Of Recall

The assessment of recall statistics is explained in table.5 and fig.11. From the entire progress assessment, the proficiency of projected scheme is verified.

5.5 Confidential measure

Preserving the wide range of security in execution medium has tend to attain high measure of confidential rate. Moreover, confidential means keeping the statistical record more secret. Here, to secure the gathered test cases any kind of check sum binary replica is used while saving the gathered test cased. Thus the assessment of confidential measure is revealed in fig.12 and table.6.

Table.6 confidential rate

fault detectio n	CBM	AG C	HP	Propose d
5	96.5	84.5	94.6	99
10	96.2	84.2	94.3	98.5
15	96	84	94	98
20	96.9	83.9	93.9	97.5
25	96.5	83.6	93.6	97

30	96.1	83.2	93.3	96.8
35	95	83	93	96.5
40	94.9	82.9	92.8	96.2
45	93	81	91	96
50	92	80	90	96.9

The highest confidential measure gained by a novel approach is 99%. Simultaneously, the model HP has gained 94.6% of confidential measure, AGC has pertained 84.5 % of confidential rate and the approach CBM has achieved 96.5% of confidential measure.

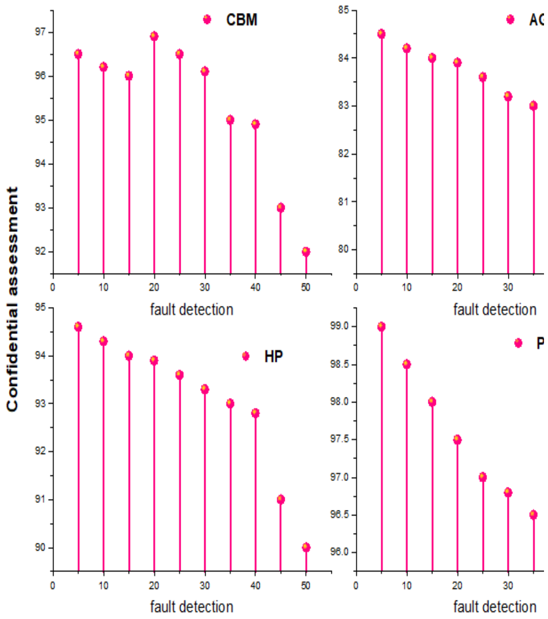

Fig.12 Confidential Assessment

From this evaluation, the developed replica has attained high confidential measure. Thus the effectives of a novel proposed replica is proved.

6. CONCLUSION

To improve the system process, here an efficient priority based regression testing replica as a novel OHGW-ALR framework is implemented using python frame. Several, test cases are processed to verify the proficient measure and running capacity of designed model. After the data training, a novel approach is designed to predict the present flaw in each test case. Before that, the gathered test cases are secured using checksum replica by performing any binary approaches. Finally, the developed replica has gained high range of exactness rate as 99 and maximum amount of processed test cases are 55. Thus, the progression assessment verified that the developed approach is sufficient to utilize in testing appliances.

REFERENCES

[1] Lity, Sascha, et al. "Retest test selection for product-line regression testing of variants and versions of variants." Journal of Systems and Software 147 (2019): 46-63.

[2] Butool, Rimsha, et al. "Improving Requirements Coverage in Test Case Prioritization for Regression Testing." 2019 22nd International Multitopic Conference (INMIC). IEEE, 2019.

[3] Dhareula, Priyanka, and Anita Ganpati. "Flower Pollination Algorithm for Test Case Prioritization in Regression Testing." ICT Analysis and Applications. Springer, Singapore, 2020. 155-167.

[4] Alshoaibi, Deema, et al. "PRICE: Detection of Performance Regression Introducing Code Changes Using Static and Dynamic Metrics." International Symposium on Search Based Software Engineering. Springer, Cham, 2019.

[5] Long, Zhenyue, et al. "WebRTS: A Dynamic Regression Test Selection Tool for Java Web Applications." 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2020.

[6] Nimară, Sergiu, et al. "Reference Images Generation for Automotive Regression Testing." 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC). IEEE, 2019.

[7] Sutar, Shantanu, et al. "Regression Test cases selection using Natural Language Processing." 2020 International Conference on Intelligent Engineering and Management (ICIEM). IEEE, 2020.

[8] Marijan, Dusica, Arnaud Gotlieb, and AbhijeetSapkota. "Neural Network Classification for Improving Continuous Regression Testing." 2020 IEEE International Conference On Artificial Intelligence Testing (AITest). IEEE, 2020.

[9] Al-Sabbagh, KhaledWalid, et al. "Selective Regression Testing based on Big Data: Comparing Feature Extraction Techniques." 2020 IEEE International

Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, 2020.

[10] Dhareula, Priyanka, and Anita Ganpati. "Flower Pollination Algorithm for Test Case Prioritization in Regression Testing." ICT Analysis and Applications. Springer, Singapore, 2020. 155-167.

[11] Lin, Xintang, et al. "Test Case Minimization for Regression Testing of Composite Service Based on Modification Impact Analysis." International Conference on Web Information Systems and Applications. Springer, Cham, 2020.

[12] Long, Zhenyue, et al. "WebRTS: A Dynamic Regression Test Selection Tool for Java Web Applications." 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2020.

[13] Sutar, Shantanu, et al. "Regression Test cases selection using Natural Language Processing." 2020 International Conference on Intelligent Engineering and Management (ICIEM). IEEE, 2020.

[14] He, Fei, Qianshan Yu, and Liming Cai. "Efficient Summary Reuse for Software Regression Verification." IEEE Transactions on Software Engineering (2020).

[15] Bin Ali, Nauman, et al. "On the search for industry-relevant regression testing research." Empirical Software Engineering 24.4 (2019): 2020-2055.

[16] Ali, Sadia, et al. "Enhanced regression testing technique for agile software development and continuous integration strategies." Software Quality Journal (2019): 1-27.

[17] Chi, Jianlei, et al. "Relation-based test case prioritization for regression testing." Journal of Systems and Software 163 (2020): 110539.

[18] Alcocer, Juan Pablo Sandoval, AlexandreBergel, and Marco Tulio Valente. "Prioritizing versions for performance regression testing: The Pharo case." Science of Computer Programming 191 (2020): 102415.

[19] Mahdieh, Mostafa, et al. "Incorporating fault-proneness estimations into coverage-based test case prioritization methods." Information and Software Technology 121 (2020): 106269.

[20] Minhas, NasirMehmood, et al. "Regression testing for large-scale embedded software development–Exploring the state of practice." Information and Software Technology 120 (2020): 106254.

[21] Wang, Hongda, et al. "Test Case Prioritization for Service-Oriented Workflow Applications: a Perspective of Modification Impact Analysis." IEEE Access (2020).

[22] Weigl, Alexander, MattiasUlbrich, and Daniel Lentzsch. "Modular regression verification for reactive systems." International Symposium on Leveraging Applications of Formal Methods. Springer, Cham, 2020.

[23] Chen, Jinfu, Weiyi Shang, and EmadShihab. "PerfJIT: Test-level Just-in-time Prediction for Performance Regression Introducing Commits." IEEE Transactions on Software Engineering (2020).

[24] Mansky, Susannah, and Elsa L. Gunter. "Safety of a Smart Classes-Used Regression Test Selection Algorithm." Electronic Notes in Theoretical Computer Science 351 (2020): 51-73.

[25] Tsionas, Mike G., and A. George Assaf. "Symbolic regression for better specification." International Journal of Hospitality Management 91 (2020): 102638.

[26] Abualigah, Laith, and Ali Diabat. "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments." Cluster Computing (2020): 1-19.