BVG Trust

J. Sci. Trans. Environ. Technov. 2019, 12(4) : 200- 204

Scientific Transactions in Environment and Technovation

Enhancing Software Testing Framework in Public Cloud

https://doi.org/10.56343/STET.116.012.004.008
http://stetjournals.com

D. Anitha[1] and M.V. Srinath[2]

[1]PG and Research Department of Computer Science, Sengamala Thayaar Educational Trust Women's College, Sundarakkottai, Mannargudi - 614 016, Tamil Nadu, India.

[1]Department of Computer Applications, Sengamala Thayaar Educational Trust Women's College, Sundarakkottai, Mannargudi - 614 016, Tamil Nadu, India.

Abstract

Software testing is the process of testing a software product to recognize the bugs for ensuring the quality of the product. The testing is a key step in the software development process. The testing ensures the level of satisfaction of the end users. It requires a massive amount of resources to achieve reliability, performance and security. As testing is a critical and an expensive task, it is better to carry out testing in the cloud environment. Cloud computing is a novel computing standard that provides major support for the software testing and development. Distributed cloud is a cloud computing technology that interconnects the nodes located in various geographic locations. Testing as a Service (TaaS) is offered by the cloud providers as business occasion to test the cloud based applications. It allows the testers, developers and mangers to enhance the performance of testing with the help of web browsers. In this research work, a secure software testing framework with weight-based prioritization technique is proposed for performing software testing in the distributed cloud environment. At first, the test case dataset is initialized then the frequent test case in the dataset is estimated. The weight of the identical test cases is determined using the cosine similarity, then the weight based prioritization technique is applied for prioritizing the test cases. Generally, the test cases are clustered using agglomerative clustering and K-means clustering algorithms. In agglomerative clustering, the distance between two clusters are always greater and it is sensitive to noise and outliers. In K-means clustering it is difficult to predict the K-value as it has fixed number of clusters. Hence, to address the issues in the existing clustering algorithms, a cosine similarity based K-Medoid algorithm is proposed. The K-Medoid algorithm constructs a similarity matrix for clustering the test cases with similar values. As the test case clusters are similar, the execution of the test case provides accurate results. After prioritization, the K-Medoid clustering algorithm is deployed for clustering the similar test cases. As security is a major concern in cloud infrastructure, it is necessary to protect the test cases using cryptographic algorithms. In common, cryptographic algorithms including AES and DES are applied to protect the data from cloud environment attacks. But, the level of security provided by these algorithms is not satisfactory. Hence, to enhance the security of the data, an efficient Attribute key based Elliptic Curve Cryptography (ECC) algorithm is proposed for the encryption and decryption. The security level is increase by using key pair based cryptography calculation. The generated keys are maintained in the hash table for faster retrieval. Both the encryption and decryption processes exploit the same key for encrypting and decrypting the clustered test cases. The implementation of the test cases exploits a cache memory. Failure in the test case execution removes the test case from the cache memory for optimizing the memory space. The results from all the nodes in the distributed cloud environment are merged for providing the final test report.

Key words: Elliptic curve cryptography, K-means clustering, Software testing, TaaS.

Received : March 2018 Revised and Accepted : January 2019

INTRODUCTION

Software fault is defined as a condition that does not allow the software to perform its expected function. Some of the common software faults are syntax, algorithmic, computation, documentation, hardware, system software and stress etc. Software testing is defined as the process of detecting the faults during the software development process. As the presence of faults affects the software quality, the software testing and fault prediction are mandatory. After the generation of a high-quality software, it is validated for faults. If the software is detected to be faulty, suitable measures are adopted for correcting the faulty module. Whereas, when the software is non-faulty, the next processes are executed. Software testing demands the resources such as application performance, reliability, security, functionality and speed Narula(2014). Thus, to satisfy these demands, the distributed cloud computing environment is used. Exploiting the cloud

*Corresponding Author :
email: *sri_indhuja@rediffmail.com*

for software testing and maintenance has the following advantages,

- Increased security
- Reduced cost requirement for testing process
- Enhanced testing efficiency
- Optimal performance testing

Parthiban and Karthik (2014) proposed a dependency estimation based test case prioritization technique for determining the dependencies between the test cases. When compared to the random and untreated test cases, the prioritized test cases provided optimal results for the metrics such as precision, recall, F-Measure and accuracy. Nguyen, *et al.* (2011) suggested a hierarchical agglomerative clustering algorithm for enhancing the execution efficiency of the tasks in distributed cloud environment. The key advantage of the proposed algorithm was minimal execution time. But, its efficiency in integrating the map-reduce framework was not discussed. Li, *et al.* (2013) proposed a patient-centric framework for providing an efficient data access control. The Attribute-Based Encryption (ABE) technique is used for encrypting the Personal Health Record (PHR). Experimental results proved that the proposed framework enhanced the security, scalability and efficiency.

Rewagad and Pawar (2013) enhanced the confidentiality of the cloud data using digital signature and Diffie-Hellman key exchange integrated with Advanced Encryption Standard (AES). The Diffie-Hellman key exchange algorithm protected the data even if it was hacked. Malik and Kumar(2015) proposed a data protection model for protecting the data from cloud environment attackers. The data is encrypted using AES algorithm and authenticated using the Diffie-Hellman algorithm. The suggested model enhanced the confidentiality and security. Further, the overhead for the key computation was reduced. Ma, *et al.* (2015) suggested a Distributive Big Data Clustering algorithm based on Local Remote Coordination (DBDC-LRC) for minimizing the transmission cost and enhancing the clustering accuracy. The suggested approach provided increased accuracy, increased robustness and minimal cost requirement.

Definition of the Problem

Software testing is the periodic monitoring of reliability, security, speed, application performance and functionality of the software. The increase in the complexity of the business applications demands an efficient testing facility. Thus, to satisfy these requirements, the cloud testing is preferred. The test case is a document that contains the test data, pre-conditions, anticipated outcome and post conditions. The similar test cases are clustered for enhancing the speed of the test case execution. Some of the commonly used clustering algorithms are agglomerative clustering and K-Means clustering. But, these algorithms do not provide an optimal clustering performance because the agglomerative clustering has the demerits such as increased distance between two clusters, sensitive to noise and inability to handle variable sized clusters. Further, the K-Means algorithm finds difficult to predict the K-value. As the cloud environment is prone to various security threats, the cryptographic algorithms such as Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are commonly used for securing the test cases. But, these algorithms are not optimal in securing the test cases.

Thus, to enhance the security of the test cases, an optimal cryptographic algorithm is mandatory. The error logs created during the execution of the test cases are exploited for providing an efficient software fault prediction. As the error logs are obtained from multiple users, the fault prediction becomes difficult. Thus, to address the issues in the existing approaches, an efficient weight- based prioritization technique with distance based K-Means algorithm is proposed. At first, the suggested approach initializes the test case dataset then the weight-based prioritization technique is suggested for estimating the support value of all the test cases. With the estimated weight values, the test cases are prioritized. The K-Medoid algorithm clusters the test cases with an objective of minimizing the target function. The test case clusters are protected from various security threats using Attribute-based ECC algorithm. The memory space is optimized by removing the failed test cases from cache memory. While executing the test cases, each user may encounter different errors. Thus, the FSS algorithm is proposed for removing the irrelevant features from the error logs then the DKM algorithm is used for clustering the error logs based on the faults that are responsible for the error.

Scope of Research Work

This research work mainly aims at providing a secure software testing in the distributed cloud environment. The key objectives of this research work are as follows,

- To prioritize the test cases using weight-based prioritization technique.
- To implement the cosine similarity based K-Medoid clustering algorithm for clustering the prioritized test cases.
- To enhance the security of the test case execution using attribute key based Elliptic Curve Cryptography (ECC) algorithm.

• To reduce the irrelevant data using Feature Subset Selection (FSS) algorithm

• To implement the Distance based K-Means (DKM) algorithm for clustering the big data.

Methodology

Software testing is the key phase involved in the software development life cycle. To minimize the cost of the testing process, the test cases are prioritized. There exists multiple techniques such as Dependency Structure Prioritization (Nivethitha & Sriram, 2013), fuzzy expert system (C. Hettiarachchi, 2016), modified Ant Colony Optimization (m-ACO) (K. Solanki, 2015) for prioritizing the test cases, but these techniques consume more execution time for the prioritization process. Further, it consumes more computation time for the encryption and decryption processes. Thus, to overcome these limitations, an efficient weight based prioritization technique is suggested. Initially, a test case dataset is initialized,the distributed cloud environment is created with 'N' number of nodes then the test case dataset 'D' is initialized. Every test case in the dataset 'D' is allocated a test case ID.Then the similarity between the test cases are estimated using the cosine similarity for generating an adjacency matrix. Based on the generated matrix, the weight based prioritization technique prioritizes the test cases. Finally, the similar test cases are clustered using the K-Medoid clustering algorithm. To enhance the security, an elliptic curve cryptography is suggested. Further, by eliminating the failed test cases from the cache memory, the memory optimization is achieved. When the software testing is executed as a SaaS in cloud, it may encounter various faults. Thus, to cluster these faults, the distance based K-Means clustering algorithm is proposed.

Original Contributions

Contribution 1: A weight based prioritization technique for secure software testing in distributed cloud environment

Software testing is defined as the process of periodic monitoring of faults that occur during the software development. An efficient software testing demands the resources such as increased performance, reliability, security, functionality and speed. Thus, to meet all these demands a distributed cloud environment is deployed. One of the important techniques used for testing is test case prioritization. It prioritizes the test cases based on the factors such as importance, functionality and business impact, etc. Multiple techniques like information retrieval based prioritization, dependency structure prioritization and fuzzy expert system are used in the existing for prioritizing the test cases. But, these techniques consume more execution time and memory consumption for the prioritization process. Thus, to address these issues, an efficient software testing framework with weight-based prioritization technique is proposed. The suggested framework initializes the test case and determines the most frequent test case in the dataset. The weight of the similar test cases is determined using cosine similarity. Based on the estimated weight value, the proposed weight based prioritization technique prioritizes the test cases. After the prioritization process, the K-Medoid algorithm clusters the similar test cases. The securities of the test cases are enhanced by the attribute key-based Elliptic Curve Cryptography (ECC) algorithm. Further, the memory consumption of the test case execution is reduced by using the cache memory.

Initially, the test case dataset 'D' is initialized, then for every test case in the dataset, a test case ID is allocated. The similarity between the test cases is estimated using cosine similarity as follows,

$$\cos\theta = \frac{\sup(T_i)*\sup(T_j)}{|\sup(T_i)|.|\sup(T_j)|} \quad (1)$$

Where, T_i and T_j are the test case specifications in dataset 'D'.With the estimated cosine similarity values, an adjacency matrix is constructed. The mean of the similarity values is estimated for determining the weight of the test cases. Based on the weight value, the test cases are prioritized using weight based prioritization technique as follows,

Sort $(T_i \in D)$

For each Test case $T_i \in D$

Weight (T_i) =Find mean (T_i, D) (2)

End for

Apply priority based on Weight value

The number of nodes in the distributed cloud environment is initialized as the number of clusters, then based on the number of nodes, the test cases in

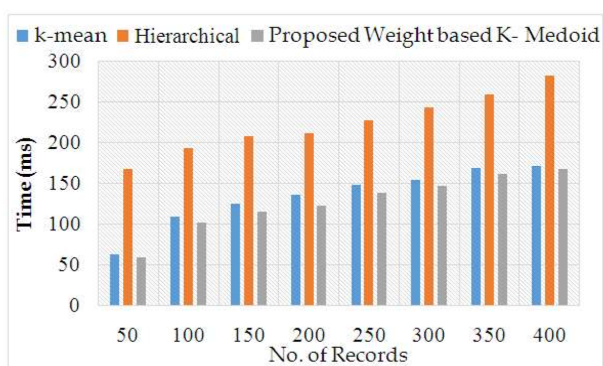

Fig. 1. Comparison of execution time for the existing and proposed methods

the adjacency matrix are clustered using K-Medoid algorithm. The key objective of the proposed algorithm is to minimize the target function represented as follows,

$$\sum_{a=1}^{k} \sum_{b \in L_i} d(b, j_b) \quad (3)$$

Where, k denotes the number of clusters, L_i denotes the cluster and $d(b, j_b)$ represents the target function. The test cases are prevented from the security threats of the cloud using Attribute-key based ECC algorithm. The suggested algorithm assigns the node ID of a node as its attribute key. The generated attribute key is maintained in the hash table with the corresponding node information. With the generated attribute key, the ECC algorithm encrypts and decrypts the clusters. Both the encryption and decryption processes consume the same private key and public key. The execution of the test cases is based on the cache memory. When a test case gets failed in their execution, it is eliminated from the cache memory for optimizing the memory space.The analysis of execution time for the existing K-Mean, hierarchical clustering, and the proposed weight based K-Medoid algorithm is depicted in Fig. 1. The analysis results prove that the proposed weight based prioritization method provides minimal execution time than the existing algorithm.

Contribution 2: Clustering the failure logs using Feature Subset Selection-Distance based K-Means (FSS-DKM) algorithm

When the cloud users execute the test cases as a Software as a Service (SaaS) they may encounter various faults. Thus, to provide an efficient software fault prediction, these faults should be categorized. But, as the error logs are provided by various users, the size of the error logs is generally big. Thus, to address this issue, an efficient Feature Subset Selection-Distance based K-means (FSS-DKM) algorithm is proposed. The FSS algorithm is used for preventing the redundant and irrelevant features. After eliminating the irrelevant data, the remaining features are clustered using the distance based K-Means algorithm. The suggested algorithm estimates the distance between every object and every cluster for allocating the object to the nearest cluster then estimates the mean for every cluster to update the cluster center. The suggested approaches are implemented and their results are compared with the results of the existing approaches.

As the irrelevant and redundant features affect the accuracy and efficiency of the clustering process, an FSS algorithm is proposed. The FSS algorithm exploits a bottom-up hierarchical clustering approach for clustering the features into groups. At first, the similarity between every pair of features are estimated then based on the similarity values, the features are clustered into a single cluster. The same process is repeated till the similarity between two clusters is less than the threshold 'θ'.

$$SU(X,Y) = \frac{2\,Gain(X|Y)}{H(X)+H(Y)} \quad (4)$$

After clustering the features, the similarity between the clusters are computed and the similar clusters are combined into a single cluster. The same process is repeated till the similarity between two clusters is less than the threshold 'è'. After eliminating the irrelevant features, the remaining features are clustered using distance based K-means clustering algorithm. The suggested algorithm includes two major steps such as estimation of the distance between every object and every cluster for allocating the object to the nearest cluster and estimation of the mean for every cluster to update the cluster centers.

CONCLUSION

In this research work, the test cases in the distributed cloud environment are tested by the clients and the error logs of all the test cases are clustered for making the suitable software fault prediction. In the initial phase of this research work, the test cases are prioritized using weight-based prioritization technique and executed in the distributed cloud environment. To validate the performance of the suggested technique, it is compared with the existing K-Mean algorithm, Hierarchical algorithm, TCP, ART, LBM and greedy approaches for the metrics such as percentage of defect detected, percentage evaluation of test case, entropy, execution time, test case execution time, prioritization time, user acceptance testing, and scalability testing. The comparison results show that the suggested algorithm provides optimal results for all the metrics than the existing techniques. In the second phase of this research work, the irrelevant error features are eliminated using the proposed FSS algorithm. The distance based K-Means algorithm clusters the test cases based on the failure behavior. To validate the performance of the proposed FSS-DKM approach, it is compared with the existing PDPCM algorithm for the metrics such as time consumption, error rate, Adjusted Rand Index (ARI). When compared to the existing PDPCM algorithm, the suggested FSS-DKM approach provides optimal results for all the metrics.

REFERENCES

Hettiarachchi, C., Do, H. and Choi, B. 2016. Risk-based test case prioritization using a fuzzy expert system. *Information and Software Technology,* 69: 1-15. https://doi.org/10.1016/j.infsof.2015.08.008

Li, M., Yu, S., Zheng, Y., Ren, K., and Lou, W. 2013. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems*, 24(1): 131-143. https://doi.org/10.1109/TPDS.2012.97

Ma, C., Liang, X., and Ma, Y. 2015. *A Succinct Distributive Big Data Clustering Algorithm Based on Local-Remote Coordination*. Paper presented at the IEEE International Conference on Systems, Man, and Cybernetics (SMC). PMid:25521795 https://doi.org/10.1109/SMC.2015.322

Malik, R. and Kumar, P. 2015. Cloud Computing Security Improvement using Diffie Hellman and AES. *International Journal of Computer Applications*, 118(1): 25-28. https://doi.org/10.5120/20711-3030

Narula, G. S. T. 2014. Framework for Analyzing and Testing Cloud based Applications. *International Journal of Advanced Research in Computer Science and Software Engineering, 4:* 592-596.

Nguyen, C. D., Marchetto, A. and Tonella, P. 2011. *Test case prioritization for audit testing of evolving web services using information retrieval techniques.* Paper presented at the IEEE International Conference on Web Services (ICWS). https://doi.org/10.1109/ICWS.2011.75 PMid:22138299

Nivethitha, S. and Sriram, V. S. 2013. Consolidating Batch and Transactional Workloads Using Dependency Structure Prioritization. *International Journal of Engineering and Technology (IJET), 5:* 1328-1334.

Parthiban, S. and Karthik, D. 2014. Establishing a Test Case Prioritization Technique Using Dependency Estimation of Functional Requirement. *International Journal of Innovative Research in Science, Engineering and Technology, 3*: 1526-1529.

Rewagad, P. and Pawar, Y. 2013. *Use of digital signature with diffie hellman key exchange and AES encryption algorithm to enhance data security in cloud computing.* Paper presented at the International Conference on Communication Systems and Network Technologies (CSNT). https://doi.org/10.1109/CSNT.2013.97

Solanki, K., Singh, Y. and Dalal, S. 2015. Test case prioritization: An approach based on modified ant colony optimization (m-ACO). *International Conference on Computer, Communication and Control (IC4)*, P. 1-6. https://doi.org/10.1109/IC4.2015.7375627 PMid:27512541 PMCid:PMC4959403