International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AN OPTIMIZED MUTATION TESTING USING HYBRID METAHEURISTIC TECHNIQUE WITH MACHINE LEARNING FOR SOFTWARE DEFECT PREDICTION

Dushyant Singh*[1] & Parulpreet Singh[2]
*1&2Baddi University of Emerging Sciences and Technology, Baddi, India

DOI: https://doi.org/10.29121/ijesrt.v10.i3.2021.10

ABSTRACT

Software defect prediction model based on the mutation testing is a pioneering method for the fault-based unit testing in which faults are detected by executing certain test data. This paper presents an Optimized Mutation Testing (OMT) technique based software defect prediction model using the concept of hybrid metaheuristic technique. Here, hybridization of OMT with Enhanced Learning-to-Rank (ELTR) is used for the feature extraction from mutation testing based data generation mechanism. In the proposed approach, first hybrid technique is used for the test data feature extraction then this data is exercised to cover all mutants present in the specific program under test and then machine learning based Random Forest as an ensemble classifier is used as a classifier. The proposed method can improve the testing as well defect prediction efficiency by deleting the redundant test data. In this research work, two models are implemented for the software defect prediction using the ELTR and LTR. At last, the performance parameters such as Detection Rate, Defect Predication Value, Execution Time, Percentage of Fault Negative Rate and Percentage of Fault Rate are measured and compared with the existing work to validate the proposed model.

KEYWORDS: Mutation testing, Software Defects, Defect Prediction, Software Quality, Machine Learning Algorithms, Random Forest, Learning-to-Rank and Enhanced Learning-to-Rank.

1. INTRODUCTION

To support testing activities in software area, mutation testing is a beneficial steps that realizes the idea of using artificial defects in Software Defect Prediction [1]. Here, the process of mutation is typically used as a way to evaluate the adequacy of test suites, to guide the generation of test cases, and to support experimentation [2-3]. Mutation has reached a maturity phase and gradually gains popularity both in academia and in traditional industry for research purpose to analyze the defects in software. Defects can be defined in a disparate ways but are generally defined as aberration from specifications or ardent expectations which might lead to failures in procedure. Software defect data analysis is of two different types such as:

- **Classification**
- **Prediction**

Both techniques can be used to extract models describing significant defect data classes or to predict future defect trends [4]. Classification predicts categorical or discrete, and unordered labels, whereas prediction models predict continuous valued functions. Such analysis can help us for providing better understanding of the software defect data at large [5-6]. This research article, we presents an Optimized Mutation Testing (OMT) techniques using the hybridization of metaheuristic technique for software defect prediction the there process is shown in the Fig. 1.

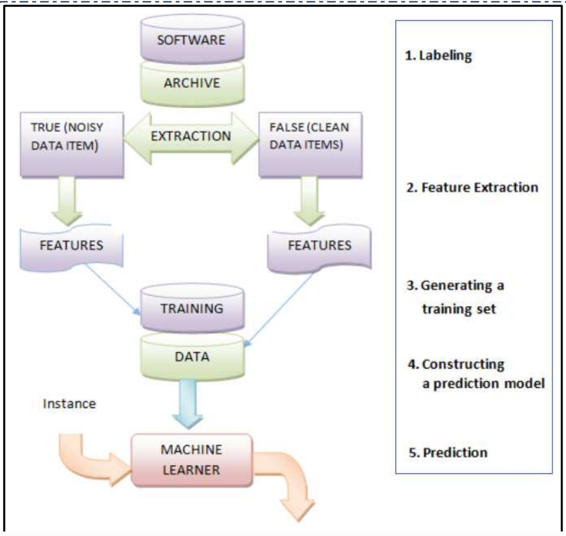

Fig. 1: Process of OMT-based Software Defect Prediction

Labeling: Defect data should be gathered for training a prediction model using the concept of OMT techniques. In this process usually extracting of instances i.e. data items from software archives and labeling (True = 1 or False = 0) is done.

Feature Extraction: This step involves extracting of features for prediction of the labels of instances using the concept of error calculation method. It can be used for training. Aspects will be extracted on the groundwork of labels of occasions. There are some points of defect predictions such as alterations, keywords and complexity metrics. This step involves extracting of points for prediction of the labels of occasions. After characteristic extraction training set can be furnished to the machine learner for setting up a prediction mannequin. General features for defect prediction are complexity metrics, keywords, changes, and structural dependencies. By combining labels and features of instances, we can produce a training set to be used by a machine learner to construct a prediction model.

Training Set Generation: After feature extraction using the concept of hybrid technique, we collect the data in terms of True and False classes and create a training set.

Prediction Model Construction: General machine learners such as Random Forest is used to build a prediction model by using a training set. The model can then obtain a new instance and predict its label, i.e. TRUE or FALSE.
Prediction: The evaluation of a prediction model requires a testing data set besides a training set. The labels of instances in the testing set are predicted and the prediction model is evaluated by comparing the prediction.

Over the past decade, related to the fundamental problems of mutation testing and sets out the challenges and open problems for the future development of the OMT techniques for software defect prediction [7]. It also collects advices on best practices related to the use of mutation in empirical studies of software testing, so, we try to design an OMT techniques using the Random Forest as ensemble approach to the extract unique set of features for training of proposed software defect prediction. There lots of research already proposed by different authors using different optimization approaches but in the existing model do not use the combination optimization technique for OMT-based software defect prediction model. So, the main motivation behind the development of this model is solve out the existing drawbacks and we introducing an OMT techniques for software defect prediction using the concept of Random Forest as a classifier and the major contributions in this research are listed as:

- Firstly, we presents a survey to analyze the existing software defect prediction techniques in recent years to find out the problems faced by researchers.
- To design an approach for the mutation analysis base on the features extraction.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

- Implement a machine learning with Random Forest classification technique as an ensemble approach for the software defect prediction model training using the extracted features.
- To validate the proposed OMT-based software defect prediction model, a comparison with the existing schemes are performed on the behalf of performance parameters like Detection Rate, Defect Predication Value, Execution Time, Percentage of Fault Negative Rate and Percentage of Fault Rate [8].

We introducing the concept of Random Forest as an ensemble classifier to design an OMT-based software defect prediction model and the remaining article is arranged as follows. In Sect. 2, related works survey are reviewed where in Sect. 3 describes the material and method of proposed model. The experimented simulation results are presented in Sect. 4 and in the Sect. 5, the conclusion with the future possibilities is presented.

2. RELATED WORK

In this section, existing work are analyzed to find out the issues related to the OMT techniques for software defect prediction because the researchers did not pay much attention toward the mutation testing approaches. In 2020, ***Y. Shao, et al.,*** had conducted a research on software defect prediction model on the basis of correlation weighted class association rule mining (CWCAR) [9]. A multi-weighted supports-based framework was leveraged instead of the conventional support-confidence approach for maintaining the class imbalance and the heuristic approach based on correlation was employed for allocating feature weight. In addition, the ranking, pruning and prediction stages were optimized on the basis of weighted support in this paper. The results demonstrated that the CWCAR outperformed existing classifiers concerning Balance, MCC and G-mean. For the future work, the associations and differentiations of class association rules would discover among different projects. Furthermore, the application of this technique would explore further in several real-world fields. ***H. GAO, et. al.***, in 2019 emphasized on providing the novel idea to predict software defect with the help of complex network features for signifying defect information [10]. First of all, the selection of eighteen versions of nine open source projects had done using certain rules. Afterward, a logistic regression model was developed on the basis of three types of attributes for the assessment of potential of the predictive defect of complex network features. It was also demonstrated that the merged features were not efficient like complex network features while predicting defect for cross-version software defect prediction. However, these features outperformed the conventional code features. In 2019, ***Z. Tian et. al.,*** analyzed that software defect prediction was very significant in software engineering. The defects which were discovered in historical software modules were employed for predicting defects in novel software modules and the decision support was provided to plan and process management in software project with it [11]. The fundamental research direction was ML in the field of AI as various disciplines had covered in it. The utilization, verification and study of different ML techniques had performed for achieving high performance in the research of software defect prediction. This paper discussed basic framework of predicting software defect on the basis of machine learning and a number of prediction models were studied and discussed in this paper. At last, the major issues related to defect prediction were reviewed and the solutions to deal with these issues had also suggested in this paper. ***Md Alamgir Kabir, et.al*** in 2019 discussed the existence of concept drift along with associated effects on performance of software defect prediction [12]. An empirical technique named DDM had implemented and the chi-square test with Yates continuity correction had employed for computing its statistical importance. The main purpose of this paper was that the concept drift was determined and the base model was standardized according to it. It was demonstrated in empirical study that the concept drift that occurred in software defect datasets and the performance of prediction models had degraded its existence subsequent manner. The chi-square test with Yates continuity correction was employed to recognize 2 kinds of concept drifts in the study of software defect datasets. It was recommended in this paper that software quality assurance teams had to consider the drift concept while developing prediction models. In 2019, ***Ying Liu, et.al*** presented a software defect prediction algorithm on the basis of enhanced BP NN that was utilized for enhancing the prediction accuracy in the effectual way that affected due to the imbalance of the category distribution of data within the project [13]. The SMOTE algorithm was employed for enhancing the data imbalance in the project and also for the maximization of minority samples. The issue of post-sampling data noise had resolved using ENN data cleaning algorithm. The four- layers BP neural network was optimized through SA algorithm so as the classification prediction model was developed on the AEEEM database. The cross validation had also carried out for the computation of the presented algorithm on AEEEM database. It was demonstrated in the results that the

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

performance of the model was enhanced in efficient way using presented algorithm to predict the unbalanced data. In 2019, ***G. Fan, et.al*** suggested a framework named DP-AM so as the benefit of the semantics and static metrics of programs were taken [14]. First of all, the vectors were extracted in the DPAM. The ASTs of programs had employed to map and word embedding for encoding of these vectors as digital vectors. These numerical vectors were utilized into RNN for learning semantic features of programs in automatic way. Then, association among these attributes was further established using self-attention mechanism. In addition, the global attention mechanism was carried out for producing major features among them. At last, these semantic features were incorporated with conventional static metrics to predict the software defect accurately. There were 7 open-source Java projects had utilized in Apache for the computation of this technique concerning F1-measure. The results of experiments had demonstrated that F1-measure was enhanced by 11% in average using DPAM than existing methods.

This research work is based on the software defect prediction. The software defect prediction has various phases which are pre-processing, feature extraction and classification. In the pre-processing phase, data which is taken as input is proposed to remove missing and redundant data from the dataset. When the dataset is pre-processed efficiently, prediction accuracy will be high. The second phase will be of feature extraction in which relationship will be established between attribute and target sets. In the last phase, the classification algorithms will be applied which can classify data into certain classes. In the previous research, the improved GA and PSO is applied for the feature extraction. The bagging method with SVM classifier is applied for the classification. It is analyzed this technique is unable to solve the problem of class unbalancing and also accuracy is low. The techniques needs to propose for the software defect prediction which give high accuracy for the prediction. Based on the above survey of existing research, we concluded and decided to present a software defect prediction model using the concept of Random Forest and Enhanced Learning-to-Rank (ELTR) as a classifier.

3. MATERIAL AND METHOD

In this section of research paper, we explain the used materials and methodology of proposed OMT-based software defect prediction model. To predict the software defects phases of pre-processing, feature extraction and classification are applied in this research work. The techniques of Random Forest with ELTR is applied for the software defect prediction in the previous research. The flowchart of proposed OMT-based software defect prediction model is shown in the Fig. 2.

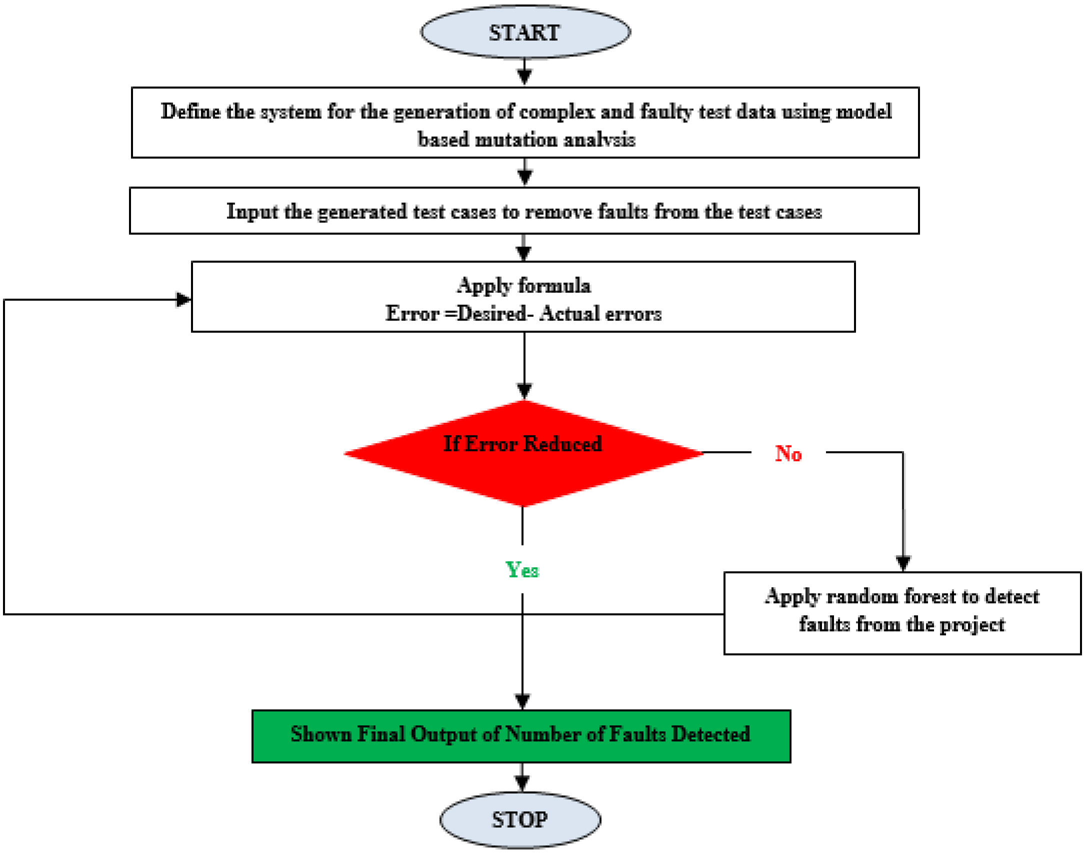

Fig. 2: OMT-based Software Defect Prediction Flowchart

In the proposed an OMT-based software defect prediction model, we presents a comparative analysis to validate the proposed ELTR model with LTR model using Random Forest. As shown in Fig. 3, the mutation testing is the type of testing and in this testing, test cases are generated which test the software. The test cases which are generated have the faults which reduce efficiency of fault prediction.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

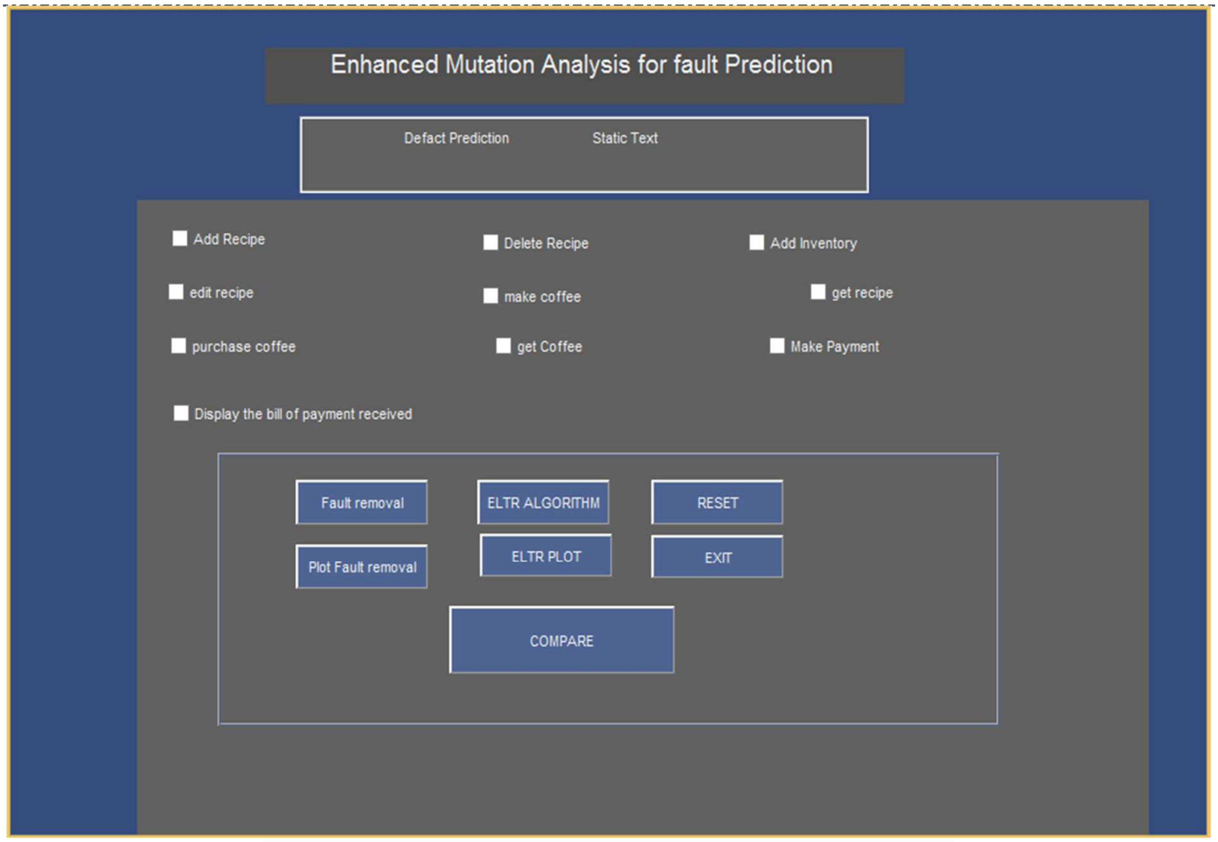

Fig. 3: Proposed OMT-based Software Defect Prediction in MATLAB

Dataset Description: Data comes from McCabe and Halstead features extractors of source code. These features were defined in the 70s in an attempt to objectively characterize code features that are associated with software quality. The nature of association is under dispute. Notes on McCabe and Halstead follow. The McCabe and Halstead measures are "module"-based where a "module" is the smallest unit of functionality. In C or Smalltalk, "modules" would be called "function" or "method" respectively.

4. RESULTS AND ANALYSIS

In this section, we describe the simulation results of the proposed OMT-based software defect prediction using ELTR and the simulation results based on the Detection Rate, Defect Predication Value, Execution Time, Percentage of Fault Negative Rate and Percentage of Fault Rate is given below with comparison using with other existing LTR technique.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

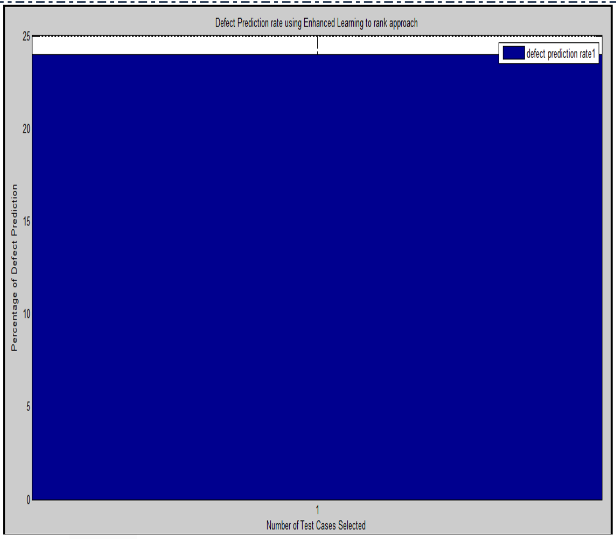

Fig. 4: Percentage of defect prediction of RLTR

Fig. 4 shows defect prediction values along with number of iterations. The algorithm of mutation analysis is applied which predict the faults from the test cases. The iterations are executed which will detect the faults at each iteration of case.

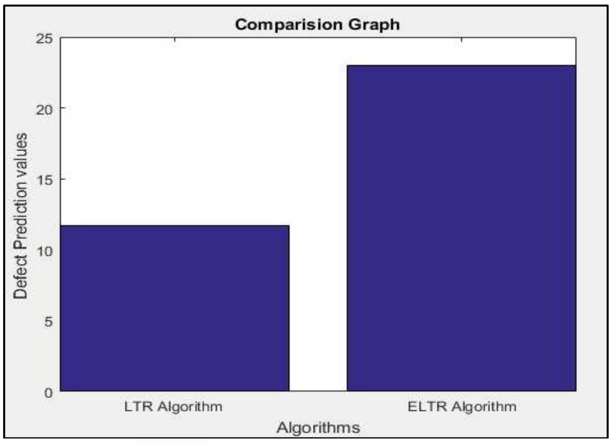

Fig. 5: Defect Prediction Analysis

Fig. 5 shows the comparison of LTR and ELTR values to analyze their performances. The ELTR approach has high defect prediction value as compared LTR algorithm.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

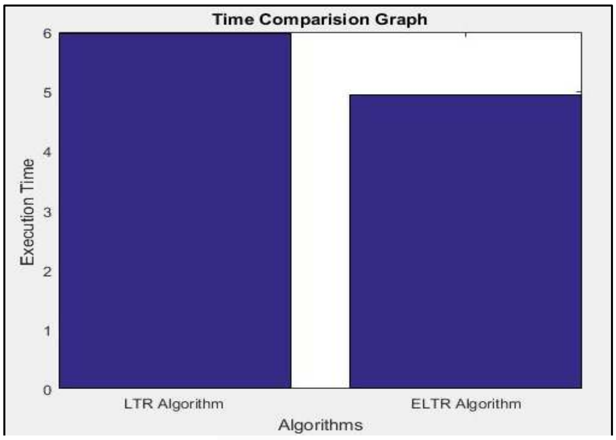

Fig. 6: Time Analysis

Fig. 6 shows the comparison of Mutation and ELTR values to analyze their performances. The ELTR algorithm has low execution value than LTR approach.

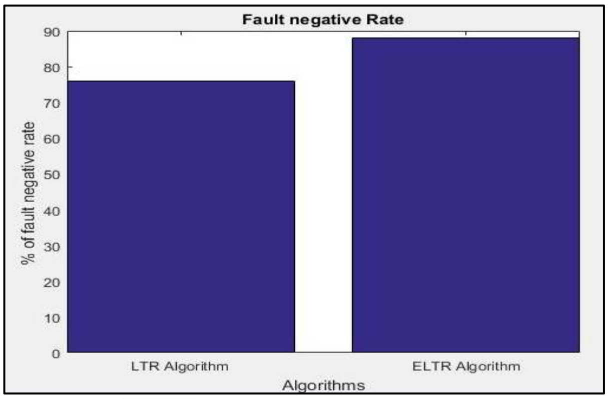

Fig. 7: Fault Negative Rate Analysis

Fig. 7 shows the comparison of Mutation and ELTR values to analyze their performances. The ELTR algorithm has high fault negative rate as compared LTR algorithm.

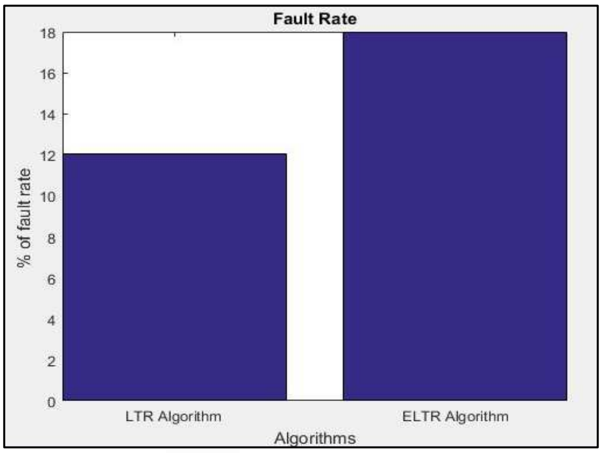

Fig. 8: Fault Rate Analysis

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Fig. 8 shows the comparison of Mutation and ELTR values to analyze their performances. The ELTR algorithm shows high rate than LRT algorithm.

Table I: Defect Prediction Values using LRT

Iteration Number	Defect Prediction Rate using Mutation Approach
1st iteration	15.773
20thiteration	12.2115
40thiteration	11.9706
60thiteration	11.9073
80th iteration	11.9073
100thiteration	11.9073
120th iteration	11.9073
140th iteration	11.9073
160th iteration	11.9073
180th iteration	11.9073
200th iteration	11.9073

As shown in the Table I, the algorithm of mutation analysis is applied which predict the faults from the test cases. The iterations are executed which will detect the faults at each iteration.

Table II: Defect Prediction Values using ELTR

Iteration Number	Defect Prediction Rate using Enhanced LTR Approach
1st iteration	29.4829
20thiteration	29.2466
40thiteration	28.6108
60thiteration	28.1757
80th iteration	24.1416
100thiteration	24.1416
120th iteration	24.1077
140th iteration	24.0198
160th iteration	24.0198
180th iteration	23.9826
200th iteration	23.9669

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Table II shows the values of faults from the test cases. It is analyzed that value of fault increase with the number of iterations. Form above simulation results based on the is show that the proposed OMT-based software defect prediction model using ELTR have its own impact with Random Forest and also show the improved results compare to the LTR.

5. CONCLUSION AND FUTURE WORK

In this paper, an OMT-based software defect prediction model is presents using the concept of hybrid metaheuristic technique. Software defect along with an intrinsic element of software product, is also an important aspect of software quality. Software defects are an unavoidable co product of the developed software. In addition to this, the guarantee of software quality assurance is not so easy and requires a lot of time too. There are different ways to define defects, such as in terms of quality. However, the defects are generally defined in the form of deviations from specifications or expectations which may be the reason of failure in functioning. In general, various software projects do not have sufficient time and workforce available for eliminating all the defects prior to the release of a specific product. This may affect the overall product quality and probably the status of an organization that deliver the product. In this research work, two different model using LTR and ELTR are implemented for the software defect prediction and also compared to identify the effectiveness of proposed OMT-based model using ELTR. A testing method called fault detection is implemented for detecting faults within the software or within the input test cases. For the faults in the software the mutation is known as the algorithm. Reduction in the fault detection rate is the algorithm that chose the population randomly. From the previous experience and driving new values, back propagation algorithm is implemented from where the system learns in this work. For testing the procedure a test case is a set of procedure. Test case is considered as a condition set. The test case is used to determine the functioning of software system. The software testing is performed using test case. The software performance in normal condition was verified by designing positive test cases in severe circumstance. The software is validated by designing negative test case. An inappropriate implementation order may disclose various faults belatedly. This can cause interruption in bug removal process. The software delivery also gets affected by this. Improvement in the defect detection rate and reduction in the execution time are led by this. O the basis of the bio-inspired techniques for the fault detection rate, technique will be proposed in future. This algorithm learns from the past experience and determines novel values. Total 10 test cases are used for software testing. The simulation is implemented in MATLAB software. The outcomes of simulation depict improved fault detection rate and decreased execution time. The proposed OMT-based software defect prediction model can be compared with other models of software defect prediction to analyze reliability and also, can be further improvement using the concept of supervised learning techniques.

REFERENCES

[1]. LameesAlhazzaa, Anneliese Amschler Andrews, "Trade-Offs between Early Software Defect Prediction versus Prediction Accuracy", 2019, International Conference on Computational Science and Computational Intelligence (CSCI)
[2]. SukmawatiAnggraeni Putri, Frieyadie, "Combining integreted sampling technique with feature selection for software defect prediction", 2017, 5th International Conference on Cyber and IT Service Management (CITSM)
[3]. Yuan Chen, Xiang-heng Shen, Peng Du, Bing Ge, "Research on software defect prediction based on data mining", 2010, The 2nd International Conference on Computer and Automation Engineering (ICCAE), Volume: 1
[4]. MoloukMishmastNehi, Zahra Fakhrpoor, Mohammad R. Moosavi, "Defects in The Next Release; Software Defect Prediction Based on Source Code Versions", 2018, Electrical Engineering (ICEE), Iranian Conference on
[5]. Ning Li, Martin Shepperd, Yuchen Guo, "A systematic review of unsupervised learning techniques for software defect prediction", 2020, Information and Software Technology, Volume 122, Article 106287
[6]. Hua Wei, Changzhen Hu, Shiyou Chen, Yuan Xue, Quanxin Zhang, "Establishing a software defect prediction model via effective dimension reduction", 2019, Information Sciences, Volume 477, Pages 399-409
[7].

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[8]. Yuchen Guo, Martin Shepperd, Ning Li, "Poster: Bridging Effort-Aware Prediction and Strong Classification - A Just-in-Time Software Defect Prediction Study", 2018, IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)

[9]. Kwabena EboBennin, Jacky Keung, PassakornPhannachitta, AkitoMonden, Solomon Mensah, "[Journal First] MAHAKIL: Diversity Based Oversampling Approach to Alleviate the Class Imbalance Issue in Software Defect Prediction", 2018, IEEE/ACM 40th International Conference on Software Engineering (ICSE)

[10]. Yuanxun Shao, Bin Liu, Shihai Wang, Guoqi Li, "Software defect prediction based on correlation weighted class association rule mining", 2020, Knowledge-Based Systems, Volume 19621, Article 105742

[11]. Houleng Gao, Minyan Lu, Cong Pan, Biao Xu, "Empirical Study: Are Complex Network Features Suitable for Cross-Version Software Defect Prediction?", 2019, IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)

[12]. Zhang Tian, Jing Xiang, Sun Zhenxiao, Zhang Yi, Yan Yunqiang, "Software Defect Prediction based on Machine Learning Algorithms", 2019, IEEE 5th International Conference on Computer and Communications (ICCC)

[13]. Md Alamgir Kabir, Jacky W. Keung, Kwabena E. Benniny, Miao Zhang, "Assessing the Significant Impact of Concept Drift in Software Defect Prediction", 2019, IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)

[14]. Ying Liu, Fengli Sun, Jun Yang, Donghong Zhou, "Software Defect Prediction Model Based on Improved BP Neural Network", 2019, 6th International Conference on Dependable Systems and Their Applications (DSA)

[15]. Guisheng Fan, XuyangDiao, Huiqun Yu, Kang Yang, Liqiong Chen, "Deep Semantic Feature Learning with Embedded Static Metrics for Software Defect Prediction", 2019, 26th Asia-Pacific Software Engineering Conference (APSEC)

[16]. Yan Zhou, Chun Shan, Shiyou Sun, Shengjun Wei, Sicong Zhang, "Software Defect Prediction Model Based On KPCA-SVM", 2019, IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation.

[17]. Fuqun Huang, Lorenzo Strigini, "Predicting Software Defects Based on Cognitive Error Theories", 2018, IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW).