International Journal of Art & Humanity Science(IJAHS)
www.ijahs.com Volume 3 Issue 6, (Nov-Dec 2016), PP. 20-27

An AI Approach for Software Defect Prediction –A Review of Literature

Vinay Kumar Dwivedi
vinaydwived@gmail.com
Mahesh Kumar Singh Assistant Professor Bansal Institute of Engineering & Technology, Lucknow
mks.cse07@gmail.com

Abstract:
Faults in software systems continue to be a major problem [1]. High quality of software is ensured by Software reliability and Software quality assurance. Both these concepts are drawn in throughout the software development and maintenance process. The activities like the performance analysis, functional tests, quantifying time and budget along with measurement of metrics are used to ensure quality. A software bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from behaving as intended (e.g., producing an incorrect result) [2]. This paper surveys literature review of articles for the past many years in order to explore how various prediction methodologies have been developed during this period in order to take care of the issues related to software defect.
Keywords: Software Defects, software quality, assurance

1 Introduction

Since past many years organizations have been seeking answer to the question that how to forecast the quality issue of their software's before its final utilization. In order to tackle this issue many journals have given details about the metrics and statistical techniques. Software defect containing the quality issue can be understood in many ways but the most common one is the variation in the specific results or expectations which will ultimately lead to the failure of the operation.

Software defect prediction is the process of locating defective modules in software. For producing high quality software, the delivering final product should have as few defects as possible. For early detection of software defects could lead to reduced development costs and rework effort and more reliable software. Therefore the defect prediction is important to achieve software quality. Defect prediction metrics play the most important role to build a statistical prediction model. Most defect prediction metrics can be categorized into two kinds: code metrics and process metrics. The prediction models can then be used by the software organizations during the early phases of software development to identify defect modules. The software organizations can use this subset of metrics amongst the available large set of software metrics. These metrics can be used in developing the defect prediction models. Many researchers have used various methods to establish the relationship between the static code metrics and defect prediction. These methods include the traditional statistical methods

such as logistic regression and the machine learning methods such as Decision trees, Naive Bayes, Support Vector Machines, Artificial Neural Networks This paper provides a *critical* review of the various work carried out in this field with the purpose of identifying future avenues of research.

2. Review of Literature

Ahmet Okutan, et.al.(2012), proposed a novel method using Bayesian networks to explore the relationships among software metrics and defect proneness. Nine data sets from Promise data repository has been used and show that RFC, LOC, and LOCQ are more effective on defect proneness. In addition to the metrics used in Promise data repository, two more metrics, i.e. NOD for the number of developers and LOCQ for the source code quality has been proposed. At the end of modelling, the usefulness of the marginal defect proneness probability of the whole software system, the set of most effective metrics, and the influential relationships among metrics and defectiveness has been deduced.

Mrinal Singh Rawat et. al.(2012), identified causative factors which in turn suggest the remedies to improve software quality and productivity. They showed how the various defect prediction models are implemented resulting in reduced magnitude of defects. They presented the use of various machine learning techniques for the software fault prediction problem. The unfussiness, ease in model calibration, user acceptance and prediction accuracy of these quality estimation techniques demonstrate its practical and applicative magnetism. These modeling systems can be used to achieve timely fault predictions for software components presently under development, providing valuable insights into their quality. The software quality assurance team can then utilize the predictions to use available resources for obtaining cost effective reliability enhancements.

Supreet Kaur, et.al. (2012), studied the performance of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is evaluated for Fault prediction in Java based Object Oriented Software systems and C++ language based software components. Here, the metric based approach is used for prediction. In case of Java based dataset named as KC3, first, thirty nine metrics are used and later the worth of a subset of attributes is calculated and the number of metrics are reduced to eight. When the worth of an attribute by computing the value of the chi-squared statistic with respect to the class is evaluated, it was seen that that the BRANCH_COUNT and maxHALSTEAD_VOLUME are highest rank metrics for fault prediction in case of Java and C++ based fault prediction dataset respectively.

Karpagavadivu.K, et.al. (2012) analyzed the performance of various techniques used in software fault prediction. And also described some algorithms and its uses. They found that the aim of the fault prone module prediction using data mining is to improve the quality of software development process. By using this technique, software manager effectively allocate resources. The overall error rates of all techniques are compared and the advantages of all methods were analyzed.

Yajnaseni Dash, Sanjay Kumar Dubey, (2012) aimed to survey various research methodologies proposed to predict quality of OO metrics by using neural network approach. The application of artificial neural networks is an efficient method to estimate maintainability in object oriented system. It was seen that among the different soft computing techniques ANN possesses advantages of predicting the software maintenance effort by minimal computation. It can be used as a predictive model because of its incredible representation techniques and ability to perform complicated functions.

Ms. Puneet Jai Kaur, Ms. Pallavi, (2013) discussed data mining techniques that are association mining, classification and clustering for software defect prediction. This helps the developers to detect software defects and correct them. Unsupervised techniques may be used for defect prediction in software modules, more so in those cases where defect labels are not available.

K.Venkata Subba Reddy and Dr.B.Raveendra Babu, (2013), proposed a software reliability growth model, which relatively early in the testing and debugging phase, provides accurate parameters estimation, gives a very good failure behaviour prediction and enable software developers to predict when to conclude testing, release the software and avoid over testing in order to cut the cost during the development and the maintenance of the software. Two real world experimental data previously analyzed have been used to compare the proposed Early Estimation Logistic Model effectiveness with several pre-existing models.

Romi Satria Wahono and Nanna Suryana, (2013), proposed the combination of particle swarm optimization and bagging technique for improving the accuracy of the software defect prediction. Particle swarm optimization is applied to deal with the feature selection, and bagging technique is employed to deal with the class imbalance problem. The proposed method is evaluated using the data sets from NASA metric data repository. Results have indicated that the proposed method makes an impressive improvement in prediction performance for most classifiers.

Ahmet Okutan1 and Olcay Taner Yıldız, (2013) proposed a new kernel method to predict the number of defects in the software modules (classes or files). The proposed method is

based on a pre-computed kernel matrix which is based on the similarities among the modules of the software system. Novel kernel method with existing kernels in the literature (linear and RBF kernels) has been compared and show that it achieves comparable results. Furthermore, the proposed defect prediction method is also comparable with some existing famous defect prediction methods in the literature i.e. linear regression and IBK. It was seen that prior to test phase or maintenance, developers can use the proposed method to easily predict the most defective modules in the software system and focus on them primarily rather than testing each and every module in the system. This can decrease the testing effort and the total project cost automatically.

Sonali Agarwal and Divya Tomar, (2014), proposed a feature selection based Linear Twin Support Vector Machine (LSTSVM) model to predict defect prone software modules. F-score, a feature selection technique, is used to determine the significant metrics set which are prominently affecting the defect prediction in a software modules. The efficiency of predictive model could be enhanced with reduced metrics set obtained after feature selection and further used to identify defective modules in a given set of inputs. They evaluated the performance of proposed model and compared it against other existing machine learning models. The experiment has been performed on four PROMISE software engineering repository datasets. The experimental results indicate the effectiveness of the proposed feature selection based LSTSVM predictive model on the basis standard performance evaluation parameters.

Mohamad Mahdi Askari and Vahid Khatibi Bardsiri (2014) used multilayer neural network method in order to improve and increase generalization capability of learning algorithm in predicting software defects. In order to solve the existing problems, a new method is proposed by developing new learning methods based on support vector machine principles and using evolutionary algorithms. The proposed method prevents from overfitting issue and maximizes classification margin. Efficiency of the proposed algorithm has been validated against 11 machine learning models and statistical methods within 3 NASA datasets. Results reveal that the proposed algorithm provides higher accuracy and precision compared to the other models.

Mrs.Agasta Adline, Ramachandran. M(2014) Predicting the fault-proneness of program modules when the fault labels for modules are unavailable is a challenging task frequently raised in the software industry. They attempted to predict the fault–proneness of a program modules when fault labels for modules are not present. Supervised techniques like Genetic algorithm based software fault prediction approach for classification has been proposed.

Xiaoxing Yang, et.al. (2014) introduced a learning-to-rank approach to construct software defect prediction models by directly optimizing the ranking performance. They built the model on previous work, and further studied whether the idea of directly optimizing the model performance measure can benefit software defect prediction model construction. The work includes two aspects: one is a novel application of the learning-to-rank approach to real-world data sets for software defect prediction, and the other is a comprehensive evaluation and comparison of the learning-to-rank method against other algorithms that have been used for predicting the order of software modules according to the predicted number of defects. Our empirical studies demonstrate the effectiveness of directly optimizing the model performance measure for the learning-to-rank approach to construct defect prediction models for the ranking task.

Pooja Paramshetti, D. A. Phalke, (2014), surveyed various machine learning techniques for software defect predication. They observed that software defect is indeed a major issue in software engineering. Software defect module prediction using different machine learning techniques is to improve the quality of software development process. By using this technique, software manager effectively allocate resources. For predicting defects they analyzed the advantages and limitation of Artificial neural network, Support vector machine, Decision tree, Association rule and Clustering machine learning techniques.

Pooja Paramshetti , D. A. Phalk, (2015), applied association rule discovery for detecting software entities that are likely to be defective in software systems. According to them this techniques is useful to evaluate software defects. When a problem arises due to the increasing complexity of a program, then solutions are being submitted by finding software defect. The main feature that distinguishes our approach from others is using a k-means and Apriori method. It is possibly the best algorithm for the software defect problem. Standard dataset have been used for experimental purpose. The focus is to improve the quality and feasibility of the software. In our scenario, the result heavily depends on the accuracy of rules generation and based on that it will predict the software defects. The results show that proposed system generating only interesting rules which is more useful for predicting defects in software.

H. S. Shukla, Deepak Kumar Verma (2015), analysed various literatures on defect prediction and drew following conclusions. They showed that defect prediction techniques vary in the types of data they require, some require little data and other requires more. Some use work product characteristics and others require defect data only. All the techniques have strengths and weaknesses depending on the quality of the inputs used for prediction. The

problem occurs during the selection of defect detection method. The choice of defect detection method depends on factors such as the artifacts, the types of defects they contain, who is doing the detection, how it is done, for what purpose, and in which activities. Factors also include which criteria govern the evaluation. These factors show that many variations must be taken into account. Defect prediction techniques are very useful in producing quality software. With the help of the defect prediction techniques one can improve the quality and reliability of the software. Also the defect prediction leads to high quality software, reduced cost, reduced maintenance, more customer satisfaction.

3. Conclusion

This paper presents a survey of various machine learning techniques for software defect predication. From the survey, it can be observed that software defect is indeed a major issue in software engineering. Software defect module prediction using different machine learning techniques is to improve the quality of software development process. The main aim is to examine the performance of different techniques in software fault prediction. The fault prediction in software is significant because it can help in directing test effort, reducing cost, and increasing quality of software and its reliability. By using this technique, software manager effectively allocate resources. The results of software defect prediction show that the performance of the models depends upon the quality of data, its nature and the accuracy of the predictor and classifier variables. It was seen that most of the work has been carried out using NASA data. This study has shown keen interest by many researchers.

REFERENCES

1. Ahmet Okutan, Olcay Taner Yıldız,(2012) “Software defect prediction using Bayesian networks”, Empir Software Eng (2014) 19:154–181 © Springer Science+Business Media, LLC.
2. Mrinal Singh Rawat, Sanjay Kumar Dubey,(2012) “Software Defect Prediction Models for Quality Improvement: A Literature Study”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, pp 288-296.
3. Supreet Kaur, and Dinesh Kumar, “Software Fault Prediction in Object Oriented Software Systems Using Density Based Clustering Approach”, International Journal of Research in Engineering and Technology (IJRET) Vol. 1 No. 2 March, 2012 ISSN: 2277-4378

4. Karpagavadivu.K, et.al. (2012), "A Survey of Different Software Fault Prediction Using Data Mining Techniques Methods", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 8, pp 1-3.*
5. Yajnaseni Dash, Sanjay Kumar Dubey, (2012), " Quality Prediction in Object Oriented System by Using ANN: A Brief Survey", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 2,, pp.1-6.
6. Ms. Puneet Jai Kaur, Ms. Pallavi, (2013), " Data Mining Techniques for Software Defect Prediction", International Journal of Software and Web Sciences (IJSWS), International Journal of Software and Web Sciences 3(1), pp. 54-57.
7. K.Venkata Subba Reddy and Dr.B.Raveendra Babu, (2013), "LOGISTIC REGRESSION APPROACH TO SOFTWARE RELIABILITY ENGINEERING WITH FAILURE PREDICTION", International Journal of Software Engineering & Applications (IJSEA), Vol.4, No.1, pp. 55-65.
8. Romi Satria Wahono and Nanna Suryana, (2013), "Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect Prediction", International Journal of Software Engineering and Its Applications Vol.7, No.5 (2013), pp.153-166.
9. Ahmet Okutan and Olcay Taner Yıldız, (2013), "A Novel Regression Method for Software Defect Prediction with Kernel Methods", ICPRAM 2013 - International Conference on Pattern Recognition Applications and Methods, pp 216-221.
10. Sonali Agarwal and Divya Tomar, (2014), " A Feature Selection Based Model for Software Defect Prediction", International Journal of Advanced Science and Technology Vol.65 (2014), pp.39-58.
11. Mohamad Mahdi Askari and Vahid Khatibi Bardsiri (2014), " Software Defect Prediction using a High Performance Neural Network", International Journal of Software Engineering and Its Applications Vol. 8, No. 12 (2014), pp. 177-188.
12. Mrs.Agasta Adline, Ramachandran. M(2014), "Predicting the Software Fault Using the Method of Genetic Algorithm", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Special Issue 2,, pp 390-398.
13. Xiaoxing Yang, et.al. (2014), IEEE TRANSACTIONS ON RELIABILITY, This article has been accepted for inclusion in a future issue of this journal.

14. Pooja Paramshetti, D. A. Phalke, (2014), " Survey on Software Defect Prediction Using Machine Learning Techniques", International Journal of Science and Research (IJSR), Volume 3 Issue 12, pp.1394-1397.
15. Pooja Paramshetti, D. A .Phalk, (2015), "Software Defect Prediction for Quality Improvement Using Hybrid Approach", *International Journal of Application or Innovation in Engineering & Management (IJAIEM),* Volume 4, Issue 6, June 2015, pp.99-104.
16. H. S. Shukla, Deepak Kumar Verma (2015), "A Review on Software Defect Prediction", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 12, pp. 4387-4394.*