Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 7 (2017) pp. 2005-2017

http://www.ripublication.com

Comparative Analysis of the Various Data Mining Techniques for Defect Prediction using the NASA MDP Datasets for Better Quality of the Software Product

Naresh.E[#1], Vijaya Kumar B.P[#2] and Sahana.P.Shankar[#3]

#1Research Scholar, Jain University, Bengaluru and Assistant Professor, Department of ISE, Ramaiah Institute of Technology, Bengaluru, Karnataka, India.

[#2]Professor and Head of the Department, Department of Information Science and Engineering, Ramaiah Institute of Technology, Bengaluru, Karnataka, India.

[#3]Mtech Student in Software Engineering, Department of Information Science and Engineering, Ramaiah Institute of Technology, Bengaluru, Karnataka, India.

Abstract

Quality of the software product has been the prime area of focus in the past decade in the IT sector and software firms. Not being just able to meet the deliverables on time and possibly quicker time is needed, but also the ability to deliver good quality software product or even better quality at the same time is of utmost importance. The time crunch due to the shorter development and release cycles suppresses the engineer's ability to incorporate enough effort for the quality assurance activities. Hence having a way in order to be able to steer the tester's effort in the right direction is of prime importance. Defect Prediction activities will be able to tell as to where the most probable defects lie in the software product. Various data mining techniques are widely used for the defect prediction process. This paper mainly focuses on the comparison of the various techniques available and an insight as to where exactly to apply what data mining technique using the NASA MDP data sets.

Keyword-Defect Prediction, Classification, Association Rule Mining, Clustering, Neural Networks.

I. INTRODUCTION

Defect or Bug is the cause for the unusual behaviour of the software product. Unusual behaviour is directly proportional to not being able to meet the customer requirements. Such a product is of not much use to the customer and is sure to incur loss to the company in terms of money and good will. Hence a lot of research is going on as to how to improve the software quality within the limited days available of the entire software development life cycle. Many ways exist for improving the overall quality of the software thus produced such as better testing techniques, complete automation of testing activities, early defect prediction activities. Several data mining techniques exist for defect prediction process. Classification based analysis, clustering, decision tree algorithms, Association rule mining, neural network based approach are some of the popular techniques to name a few.

Extensive research is currently being carried out in the area of defect prediction using the publicly made available National Aeronautical and Space Research(NASA) Metrics Data Program(MDP) defect datasets. These datasets are made available in promise repository mainly for the experimental purpose. The defect data made available by NASA IV and V MDP contains the different software metrics and the error associated with the same at module level.

This research article is being organized as follows; the literature survey from various reputed journals and conferences such as IEEE, Elsevier; consisting of detailed analysis of the works that have already been carried out in the area of defect prediction using the various techniques. Later a tabular format of the summarized results of the various papers from the Literature survey indicating the techniques used for defect prediction in each of the case and the advantages, limitations. Finally the conclusion part concludes by telling which are the widely used techniques among the different techniques for defect prediction in the recent times.

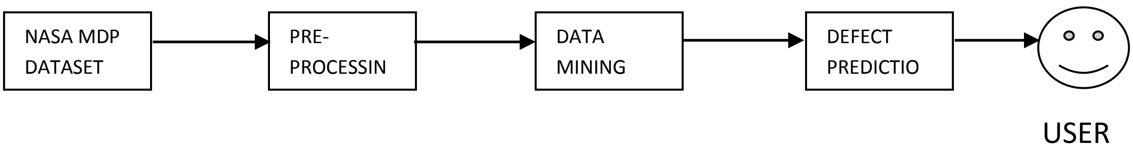
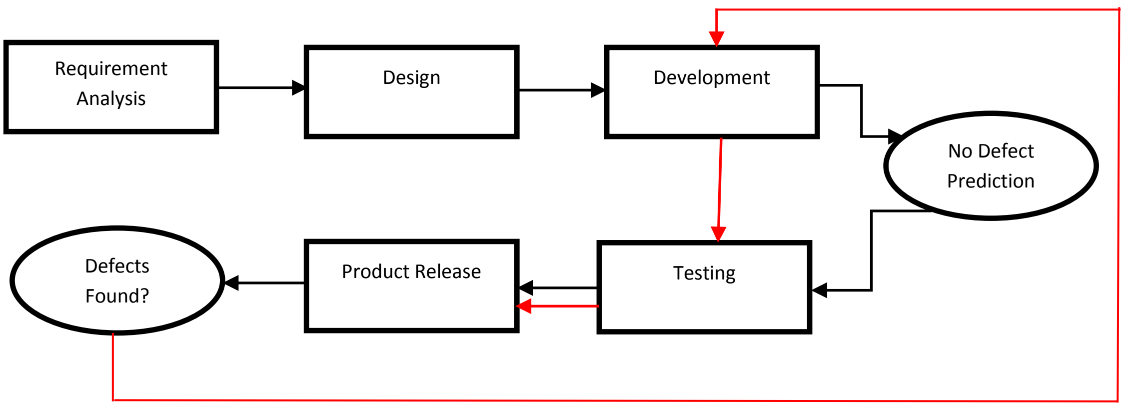


Fig.1. Generic Process of Defect Prediction

The Fig 1, depicts the generic procedure of software defect prediction process where the Datasets of the software under study are taken and is subjected to various pre-processing techniques such as normalization, elimination of redundancy, dimensionality reduction etc; Then various data mining techniques to name a few, such as clustering, classification, regression, association rule mining, neural network etc; can be applied in order to derive some knowledge out of the defect data. Understanding the results of the data mining tasks is referred to as the Defect

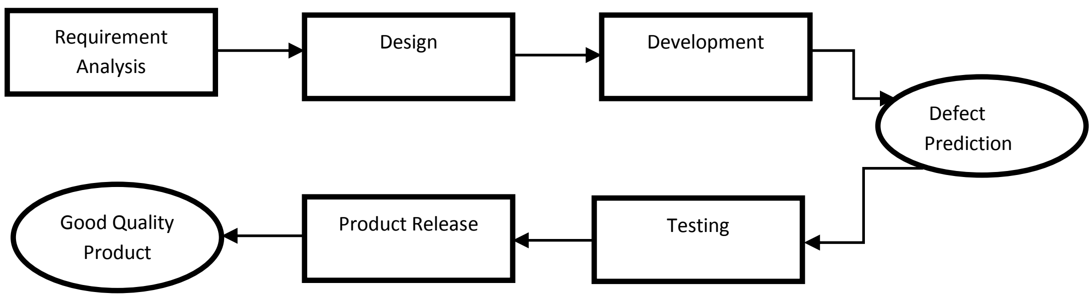
Prediction phase. The User here refers to the Manager, Team lead, Stake Holder or any responsible person who can interpret the results into a useful business process or solution.



II ROUND TESTING AND RELEASE YESREWORK

Fig.2. Traditional SDLC without Defect Prediction

The Fig 2, shows the Traditional SDLC that doesn't involve the Defect Prediction Phase. Here Requirement Analysis is followed by Design, Development, and Testing and Product Release activities. Quick and frequent agile cycles, hamper the tester's ability to carry out the testing activities completely and efficiently. Hence some parts of the product may go untested and some bugs may surpass the testing activities. In such case it results in defects being identified at the customer end, requiring rework of the development, testing and release management activities and thereby increasing the cost and effort.



I ROUND OF TESTING AND RELEASE

Fig.3. Improvised SDLC with Defect Prediction

The Fig 3, shows the improvised version of SDLC with the Defect Prediction Process as a step in-between the Development and Testing activities. With this in place, there would be no need to Re-Work as shown in Fig 2. With lesser time and effort, a better quality product can be released to the customer.

II. RELATED WORK

In [1] author proposes fuzzy integral based method for software defect prediction on mutual information. He says that the interaction among the various attributes affects the classification performance of a defect classifier. This algorithm utilizes the mutual information among different attributes to identify the fuzzy measure set function. This information reflects the attributes information and interaction information among attributes. The results show a comparative study between the popular classification algorithms such as Naive Bayesian Classifier, Logic and Libliner with the MIFI. The paper uses accuracy measures Recall, Precision and F-value for evaluation purpose. The MIFI algorithm thus proposed in this research paper can achieve better prediction effect than other three methodologies is demonstrated by experimental results which show improved results for all the accuracy measures.

In [2] author has proposed Semi-Supervised Task-Driven Dictionary Learning (STDDL) approach for defect prediction in the software modules. He has carried out the experiment on 9 of the NASA defect datasets. This approach he mainly aims at classification problem, along with a combined approach of dealing with the unlabelled datasets in a better way. The STDDL approach is compared with supervised methods such as Naïve Bayes, Coding ensemble learning (CEL), Cost-Sensitive Discriminative Dictionary Learning (CDDL) and a semi-supervised method Random Committee with Under-Sampling (ROCUS). Experimental results on accuracy measure F-value proves that CEL, CDDL, STDDL perform better than Naïve Bayes always since it fails to consider the class imbalance issue i.e. dealing with unlabelled datasets separately. Further STDDL is superior to both CEL and CDDL in tackling the class imbalance problem. Second set of results which compare STDDL with ROCUS show that STDDL outperforms ROCUS since it uses the cost-sensitive dictionary learning technique which doesn't drop any datasets from the sample for balancing. On the other hand ROCUS drops some datasets for class balancing thereby increasing the probability of dropping some important information.

In [3] author proposes a new sampling algorithm Hybrid Sapling Strategy for handling class imbalance in Defect Prediction datasets (HSDD). As seen from above, class imbalance issues needs to be resolved as a pre-processing measure before the actual defect prediction. Hence in order to improve the quality of defect prediction techniques similar to HSDD is essential. The experimental results were carried out on 10 NASA defect datasets. The experimental results proved that HSDD provides improvised results for accuracy measure G-mean when compared to oversampling algorithms such as SMOTE and Virtual. HSDD adds two new low-level metrics to the three existing datasets namely cCount and cS. G-mean is calculated for all 3 sampling algorithms using J48, Bayes, and Naïve Bayes and Random forest. The results concluded that the performance of HSDD increases with increase in the number of instances in the dataset. Hence HSDD is suitable for large-scale projects with more than 1000 instances.

In [4] author demonstrates that the proposed feature selection technique for defect prediction namely Maximal Information Coefficient with Hierarchical Agglomerative Clustering (MICHAC) is effective in producing better defect prediction results. The experiment is carried out using 11 NASA defect datasets for 4 performance measures recall, precision, F-measure and Area under the curve (AUC). MICHAC works in 2 stages, firstly selection of highly relevant features using MIC statistic followed by applying the Hierarchical Agglomerative Clustering algorithm for elimination of the redundant features. The results of comparison of MICHAC with 5 other already existing feature selection methods i.e. Chi-Square, Gain Ratio, ReliefF, TC, FECAR. The outcome of the MICHAC algorithm is applied to the existing defect prediction methods such as Naïve Bayes and Random Forest and Repeated Incremental Pruning to Produce Error Reduction (RIPPER) for comparison with existing feature selection methods. Experimental results show that MICHAC performs better in comparison with remaining 5 methods on all 3 defect prediction models.

In [5] author proves that the Improved Local Liner Embedding Support Vector Machine algorithm (ILLE-SVM) is better at detecting defects in comparison with the LLE-SVM model and SVM model alone. The drawback of the existing LLE-SVM method is that it is computationally expensive and also ignores the class imbalance problem among the datasets during the classification process. Unable to handle redundant and unbalanced datasets lead to lower accuracy of the software defect prediction process. The measures used are precision, recall, F-score and accuracy are the 4 indicators used to measure the model. The experiment carried out on 4 of the NASA defect datasets proves that ILLE-SVM model has 3%-5% higher accuracy than the LLE-SVM model.

In [6] author provides a novel method of attribute selection called Selection of Attribute with Log Filtering (SAL) which selects the best set of attributes for training the classifiers. This inturn helps to improvise the quality and performance of Software Defect Prediction. The experiments are carried out on 11 NASA defect datasets. The approach has 2 modules, first one being data pre-processing where all numeric values are discretised using $\ln(n+\in)$, where $\in$ is a small value dependent on the experiment under consideration. The second module is attribute selector where first ranking of the set of attributes is carried out and then the best ones are chosen as input for the classifier. Naïve Bayes is the classification algorithm used to compare the results with three other attribute selection features proposed in [7],[8] and [9].Balance and AUC are the evaluation metrics used. The results prove that SAL outperforms all the other datasets for almost the datasets.

In [10] author proposes a five step process for attribute selection for pre-processing of the datasets and thereby providing improved defect prediction results. The results are obtained by experimenting on 8 of the NASA defect datasets. The outcome of this is fed to the Naïve Bayes classifier which is based on conditional probability. The results are compared with two other state of art methods i.e. [11] and [12] and shows that there is an improvement of 54% in defect prediction.

In [13] author proposes a novel approach called Undersample Conditional Random Field (UCRF) for software defect prediction involving Under sampling technique and prediction. In the first stage the imbalanced nature of the datasets is being handled employing the mean shift clustering technique. Next the CRF model is being adopted that has the capacity of handling the complex features without making much changes to the above balanced dataset. The experiment is conducted on 11 of the NASA defect datasets using the performance measures Probability of Detection (PD), Probability of Non-Defect (PN) and G-mean. The results are compared with other defect prediction techniques such as CRF method without under sampling, Random CRF (RCRF), SVM and BP neural networks. Experimental results prove that UCRF technique provides an improvement of 4% over the RCRF technique. Comparing with SVM and BP Neural network there is an improvement of 3% in G-mean value.

In [14] author proposes a hybrid approach involving Locally Linear Embedding and Support Vector Machine for software defect prediction (LLE-SVM). It mainly aims to handle to issue of data redundancy in the datasets. The results of LLE-SVM is being compared with the SVM classifier on NASA defect datasets CM1. The model can be visualized in 2 stages: In the first stage, the LLE is used for dimensionality reduction. Then SVM is applied for the classification process. The performance metrics used for evaluation are F-measure, accuracy, precision and recall proved that LLE-SVM provided better results than SVM for all the measures.

In [15] author proposes a semi-supervised technique for defect prediction. The performance of semi-supervised technique is compared with a completely supervised technique such as Random Forest. In semi-supervised approach, multi-dimensional scaling is embedded as a pre-processing strategy to reduce the dimensionality of the software metrics used for prediction. The experiments are carried out on 4 of the NASA defect datasets. The performance metrics used for evaluation are AUC and PD. Results prove improved performance for all the parameters.

In [16] authors propose a Three-way decision based software defect prediction. The traditional way is the two way process where modules are classified as defect-prone or non-defect prone. But this is not favourable always especially when there is insufficient data. A third deferment decision is included and it helps decrease the misclassification cost unlike the 2-way decision making. The experiment is carried out on 11 NASA defect datasets. The performance metrics used for evaluation are F-score, accuracy and coverage (FAC). The comparison was made between 2 way decision based Naïve Bayes and 3-way decision based Naïve Bayes. Experimental results prove the efficiency of the novel approach with the other approaches.

In [17] author mainly addresses the problem of identifying the defect prone modules among the imbalanced datasets. The main problem is that the datasets tend to be dominated by the Non-Defect prone modules which inturn hinders the ability to learn the defect prone modules with much accuracy. This article proposes the Association rule based mining approach which facilitate the learning of the defective modules. The experiments are carried out on 5 NASA datasets and Recall is the performance measure used for evaluation. It has been proved that the results show an improvement of 40% in performance gain on Naïve Bayes classifier with the inclusion of this pre-processing technique when compared to execution of Naïve Bayes alone.

In [18] author proposes a hybrid approach consisting of Artificial Bee Colony (ABC) combined along with traditional Artificial Neural Network (ANN). The combined algorithm is said to be a Cost-Sensitive Neural Network where the connection weights of ANN are optimized by ABC. Experiments are conducted using 5 NASA defect datasets. The performance metrics used are accuracy, AUC, detection probability, false alarm probability, balance. Cross validation technique of N-fold was employed in order to evaluate the performance of the proposed classifier. The algorithm was compared with Naïve Bayes, C4.5, Immunos, and Artificial Immune Recognition System (AIRS). The experimental results prove a slightly improved performance in comparison with the existing classifiers, however there is no significant improvement in the performance.

In [19] author proposes a cost-sensitive defect classification technique known as CSForest; which is an ensemble of Decision Trees. While the focus of the traditional classification algorithms is optimization of accuracy, the focus of CSForest technique is cost optimization. He also proposes a technique known as CSVoting to take benefit of the decision tress in costminimization. The results are being compared with 6 other classification techniques using the 6 NASA defect datasets. The advantage of this method is that there can be extension for multiple-class cost-sensitive classification. The results are being compared with other classification algorithms such as C4.5, SVM, SysFor+Voting1[20], SysFor+Voting2 and C4.5+CSC.The performance indicators used are precision, recall, weighted precision and weighted recall for which the proposed algorithm shows better performance than the others.

In [21] authors propose a hybrid technique of feature selection and ensemble learning for the purpose of classifying defects. The ensemble learning is a novel approach that helps to overcome to problems of redundancy and data imbalance. The experiments were conducted using 3 of the NASA defect datasets. The performance indicators used for evaluation are AUC and G-mean. The results were compared with Pearson's correlation, Weighted Support Vector Machines, Random forests proved that the proposed new method is better.

III. SURVEY ANALYSIS

Table.1. Comparative Analysis from various recent publications

REFERENCES	PUBLICATION SOURCE AND YEAR	MAJOR CONTRIBUTION	NASA DATASETS USED	TECHNIQUE EMPLOYED	PERFORMANCE METRIC EVALUATED
[1]	IEEE, 2015	Considers the interaction among the attributes to improve the defect prediction quality.	CM1, JM1, KC1, PC1	Fuzzy Logic	Accuracy, Recall, Precision and F-value
[2]	IEEE, 2016	Deals with the class-imbalance problem separately and then performs classification of defects.	PC1,PC3,PC4,PC5, CM1,KC1,JM1,KC3	Semi-Supervised Learning	F-value
[3]	IEEE, 2016	Hybrid approach for dealing with class imbalance problem for very huge project with large datasets.	PC1,PC2,PC3,PC4, PC5,KC1,KC2,KC3 CM1,JM1	Sampling	G-Mean
[4]	IEEE, 2016	Applicable for the datasets that involve a lot of redundant features.	PC1,PC3,PC4,PC5, CM1,KC1,JM1,KC2.	Clustering	Precision, recall, F-measure, and AUC.
[5]	IEEE, 2015	Novel pre-processing technique to eliminate both redundancy and imbalance problem among the datasets.	PC1,PC3,PC4,CM1	Support VectorMachine	Accuracy, Recall, Precision and F-value
[6]	IEEE, 2015	Novel pre-processing technique that makes use of the traditional Naïve Bayes for classification.	PC1,PC2,PC3,PC4, PC5,CM1,KC1,JM1,KC2,KC3,KC4.	Attribute Selection and Naïve Bayes.	AUC, Balance
[10]	IEEE, 2015	The method shows a drastic improvement of 54% in defect prediction when compared with other state-of-art methods.	CM1,PC1,PC2,PC3, PC4,MC1,MW1, KC3.	Attribute Selection	Khan et al.[7] Song et al.[8]
[13]	IEEE, 2015	This technique proposes under sampling which are compared with various oversampling techniques and shows an improvement of 3%.	CM1,PC1,PC2,PC3,PC4,MC1,MC2,MW1,KC1,KC3.	Clustering and Classification.	Probability of Detection, Probability of Non-Defect and G-Mean

[14]	IEEE, 2014	This method proposed a novel way of dimensionality reduction along with handling the data redundancy problem.	CM1	Support Vector Machine	Accuracy, Recall, Precision and F-value
[15]	IEEE, 2014	This paper combines the semi-supervised learning approach along with dimensionality reduction that proves huge improvement in defect prediction.	PC1,PC3, PC4, KC1	Semi-Supervised algorithm	AUC
[16]	Elsevier, 2016	This proposed a way of classifying defects in a three way method for obtaining higher accuracy and lower decision cost.	CM1,PC1,PC2,PC3, PC4,MC1,MC2,MW1,KC1, KC3,JM1	Classification	Accuracy,F-value and coverage
[17]	Elsevier 2015	This hybrid method shows a huge improvement of 40% against the execution of Naïve Bayes alone.	CM1,MC1,KC3,PC3	Association Rule Mining	Recall
[18]	Elsevier 2015	Hybrid approach of ANN and Artificial Bee Colony is a novel technique that has been experimented to show an improvement in the performance in comparison with existing classifiers.	PC1,PC2,PC3,KC1, CM1	Artificial Neural Network	Accuracy, AUC, Probability of Detection, Probability of False Alarm, balance, NECM.
[19]	Elsevier 2015	Unlike the focus of traditional classification algorithms that is based on accuracy, the focus of this method is based on cost.	MC1,MC2,PC1,PC2, PC3,KC1	Decision Trees	Recall, Precision, Weighted Recall Weighted Precision,
[21]	Elsevier 2015	Hybrid technique of combining feature selection and ensemble learning for the purpose of defect classification.	PC4,PC2,MC1	Classification	G-Mean,AUC

Table.2. Mathematical Equations for evaluation of Metrics

S. N	PERFORMANCE METRIC	MATHEMATICAL FORMULA
1	Accuracy	Accuracy=Number of correct predictions/Total of all cases to be predicted
2	Recall	Recall=true positive/(false negative+true positive)
3	Precision	Precision= true positive/(true positive + false positive)
4	F-value	F-value= 2*((Recall*Precision)/(Recall+Precision))

5	G-Mean	G-Mean=$\sqrt{(Precision * Recall)}$
7	Probability of Detection	Detection Probability = Σ True positive/ Σ Condition Positive
8	Probability of False Alarm	False Alarm Probability= Σ False positive/ Σ Condition Negative

IV. CONCLUSION AND FUTURE WORK

It is clearly seen from the survey conducted above, that Classification Techniques have been the prime area of focus in the recent years. Much of the research work and activities in the area of Software Defect prediction is being carried out in identifying novel and hybrid techniques for classifying the modules as either defect-prone and not-defect prone. The other data mining technique such as Clustering, Association Rule mining, Fuzzy Logic, Neural Networks etc.; seem to be unexplored in comparison with the classification techniques. Also much of the research is being carried out only using the NASA MDP datasets from the PROMISE repository.

In future, focus can be shifted to other techniques of data mining for defect prediction. The datasets used could also be the other Open Source datasets that are available in the Promise Repository.

REFERENCES

[1] Fuzzy Integral Based on Mutual Information for Software Defect Prediction. Lu Liu; Kewen Li; Mingwen Shao; Wenying Liu. 2015 International Conference on Cloud Computing and Big Data (CCBD). Year: 2015. Pages: 93 - 96, DOI: 10.1109/CCBD.2015.22

[2] Semi-supervi.sed Software Defect Prediction Using Task-Driven Dictionary Learning. Ming Cheng; Guoqing Wu; Mengting Yuan; Hongyan Wan. Chinese Journal of Electronics, IEEE Conference Year: 2016, Volume: 25, Issue: 6. Pages: 1089 - 1096, DOI: 10.1049/cje.2016.08.034.

[3] HSDD: a hybrid sampling strategy for class imbalance in defect prediction data sets. M. MarufOzturk; AhmetZengin. Fifth International Conference on Future Generation Communication Technologies (FGCT), IEEE Conference. Year: 2016 Pages: 60 - 69, DOI: 10.1109/FGCT.2016.7605093.

[4] MICHAC: Defect Prediction via Feature Selection Based on Maximal Information Coefficient with Hierarchical Agglomerative Clustering. Zhou Xu; JifengXuan; Jin Liu; Xiaohui Cui2016 IEEE 23rd International

Conference on Software Analysis, Evolution, and Reengineering (SANER)Year: 2016, Volume: 1. Pages: 370 - 381, DOI: 10.1109/SANER.2016.34.

[5] Software defect prediction model based on improved LLE-SVM. Chun Shan; Hongjin Zhu; Changzhen Hu; Jing Cui; JingfengXue.2015 4th International Conference on Computer Science and Network Technology (ICCSNT). Year: 2015, Volume: 01. Pages: 530 - 535, DOI: 10.1109/ICCSNT.2015.7490804

[6] SAL: An effective method for software defect prediction. SadiaSharmin; MdRifatArefin; M. Abdullah-Al Wadud; NaushinNower; Mohammad Shoyaib. 2015 18th International Conference on Computer and Information Technology (ICCIT). Year: 2015. Pages: 184 - 189, DOI: 10.1109/ICCITechn.2015.7488065.

[7] "An attribute selection process for software defect prediction,"J. Khan, A. U. Gias, M. S. Siddik, M. H. Rahman, S. M. Khaled, M. Shoyaib et al., in Informatics, Electronics & Vision (ICIEV), 2014 International Conference on. IEEE, 2014, pp. 1–4.

[8] "A general software defect-proneness prediction framework,"Q. Song, Z. Jia, M. Shepperd, S. Ying, and S. Y. J. Liu, Software Engineering, IEEE Transactions on, vol. 37, no. 3, pp. 356–370, 2011.

[9] "Using class imbalance learning for software defect prediction," S. Wang and X. Yao, Reliability, IEEE Transactions on, vol. 62, no. 2, pp. 434– 443, 2013.

[10] Selecting best attributes for software defect prediction. PriankaMandal; Amit Seal Ami. 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE). Year: 2015. Pages: 110 - 113, DOI: 10.1109/WIECON-ECE.2015.7444011

[11] "An Attribute Selection Process for Software Defect Prediction,"J. Khan, A. U. Gias, M. S. Siddik, M. H. Rahman, S. M. Khaled, M. Shoyaib et al., In Informatics, Electronics & Vision (ICIEV), 2014 International Conference on. IEEE, 2014, pp. 1–4.

[12] "A General Software Defect-proneness Prediction Framework," Q. Song, Z. Jia, M. Shepperd, S. Ying, and S. Y. J. Liu, Software Engineering, 2011IEEE Transactions on, vol. 37, no. 3, pp. 356–370.

[13] Software Defect Prediction Based on Conditional Random Field in Imbalance Distribution. Chunhui Yang; Yan Gao; Jianwen Xiang; Lixin Liang. 2015 2nd International Symposium on Dependable Computing and Internet of Things (DCIT). Year: 2015. Pages: 67 - 71, DOI: 10.1109/DCIT.2015.21.

[14] Software defect prediction model based on LLE and SVM. Chun Shan; Boyang Chen; Changzhen Hu; JingfengXue; Ning Li. 2014 Communications Security Conference (CSC 2014). Year: 2014. Pages: 1 - 5, DOI: 10.1049/cp.2014.0749.

[15] A Semi-supervised Approach to Software Defect Prediction. Huihua Lu; BojanCukic; Mark Culp. 2014 IEEE 38th Annual Computer Software and Applications Conference. Year: 2014. Pages: 416 - 425, DOI: 10.1109/COMPSAC.2014.65.

[16] Three-way decisions based software defect prediction.Weiwei Li, Zhiqiu Huang, Qing Li. Original Research Article Elsevier 2016. Knowledge-Based Systems, Volume 91, January 2016, Pages 263-274.

[17] Improving Recall of software defect prediction models using association mining.Zeeshan Ali Rana, M. AwaisMian, ShafayShamail. Original Research Article Elsevier 2015. Knowledge-Based Systems, Volume 90, December 2015, Pages 1-13.

[18] Software defect prediction using cost-sensitive neural network. ÖmerFarukArar, KürşatAyan. Original Research Article Elsevier 2015. Applied Soft Computing, Volume 33, August 2015, Pages 263-277.

[19] Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem.Michael J. Siers, MdZahidul Islam. Original Research Article Elsevier 2015. Information Systems, Volume 51, July 2015, Pages 62-71.

[20] Knowledge discovery through sysfor: a systematically developed forest of multiple decision trees, M.Z. Islam, H. Giggins, Proceedings of the Ninth Australasian Data Mining Conference, vol. 121, Australian Computer Society, Inc., 2011, pp. 195–204.

[21] Software defect prediction using ensemble learning on selected features. Issam H. Laradji, Mohammad Alshayeb, LahouariGhouti. Original Research Article Elsevier 2015. Information and Software Technology, Volume 58, February 2015, Pages 388-402

AUTHORS PROFILE

NARESH.E received the M.Tech degree in software engineering from Ramaiah Institute of Technology, Bengaluru in 2008 and pursuing PhD from Jain University, Bengaluru under the guidance of Dr. Vijaya Kumar B. P. Currently, he is working as an Assistant Professor in the department of Information Science and Engineering at RIT, Bengaluru. His research interests include Software cost and effort estimation, empirical software engineering, and software process improvement. He is a member for ACM and Indian Society for Technical Education.

VIJAYA KUMAR B. P. received the Ph. D degree in Electrical Communication Engg., department from Indian Institute of Science (IISc), Bengaluru in 2003, M.Tech degree in Computer Science and Technology from Indian Institute of Technology, Roorke, with honors in 1992. He is currently a professor and head, in Information Science and Engg., Dept., Ramaiah Institute of Technology, Bengaluru, Karnataka, India, where he is involved in research and teaching UG and PG students, and his major area of research are Computational Intelligence applications in Mobile, Adhoc and Sensor networks. He is a senior member for IEEE.

SAHANA P SHANKAR-is currently pursuing her Masters in Software Engineering from Ramaiah Institute of Technology, Bengaluru. She has completed her B.E in Information Science and Engineering from the same college in the year 2011. She is a university rank holder-III rank in her undergraduate course. She has around 3 years of industry experience in the Software Product Testing domain from Unisys, India. She holds a U.S patent bearing patent number-US20160034382 for research and invention in software engineering domain-'Automated regression test case selector and black box test coverage tool for product testing'. She has participated in various technical paper writing contest and Tek Talks during her tenure at Unisys. She is also Awarded 'Certificate of Profession' for completing Foundation Level Certificate in Software Testing by Testing Board-ISTQB on 24th December 2011.