International Journal of Database Theory and Application
Vol.10, No.1 (2017), pp.163-174
http://dx.doi.org/10.14257/ijdta.2017.10.1.15

A Review on Software Defect Prediction Techniques Using Product Metrics

Jayanthi.R[1], Lilly florence[2] and Arti Arya[3]

[1]*Asst.Professor, MCA Department, PESIT-BSC, Karnataka*
[2]*Professor, MCA Department, Adiyamman Engg. College, Tamilnadu*
[3]*Professor, MCA Department, PESIT-BSC, Karnataka*
Email: jayanthijayashankar@gmail.com, lilly.swamy@yahoo.co.in, artiarya@pes.edu

Abstract

Presently, complexity and volume of software systems are increasing with a rapid rate. In some cases it improves performance and brings efficient outcome, but unfortunately in several situations it leads to elevated cost for testing, meaningless outcome and inferior quality, even there is no trustworthiness of the products. Fault prediction in software plays a vital role in enhancing the software excellence as well as it helps in software testing to decrease the price and time. Conventionally, to describe the difficulty and calculate the duration of the programming, software metrics can be utilized. To forecast the amount of faults in module and utilizing software metrics, an extensive investigation is performed. With the purpose of recognizing the causes which importantly enhances the fault prediction models related to product metrics, this empirical research is made. This paper visits various software metrics and suggested procedures through which software defect prediction is enhanced and also summarizes those techniques.

Keywords*: Software Defect Prediction, Software Metrics, Defect Prediction Models, Machine Learning Techniques*

1. Introduction

In the current era, as our need for complicated software has increased, the demand for excellence of software is becoming more important progressively. Now the software is being utilized almost universally and in every step of the life of human beings. The software cost such as defect & breakdown may reduce the quality of software and it creates disappointment to the customer. [1] The Primary concern of software development process is to ensure quality software at every development stage; therefore, a common goal and concern of each software development phase is to check and concentrate on improving the software quality. Software quality prediction thus aims to evaluate software quality level periodically and to indicate software quality problems early. [2] Commonly it is also called as a fault (bug) between software experts. [3] It is not so easy to manage quality software because of raising difficulties and several restrictions under which software is developed. Conversely, the software development organizations are not ready to take much risk with delivering inferior quality software. [4] Moreover it leads to disappointment among customers. The bugs are the main reason for loss of time & cost in software goods. On the other hand, gaining from the previous experience, it is possible to forecast the bugs before producing new software. To obtain this goal, we need to understand which programs are fault prone and needs optimization as per standards. From mechanisms introduced in the past, we can test the modules of the program and can enhance the defect tracking procedures that ultimately finds the cause of these bugs. Earlier research has demonstrated that entire enhancement procedure takes around 27% man hrs for only testing itself. [5] The fault forecast models are the main ingredient in

ISSN: 2005-4270 IJDTA

improving testing procedures. Even these models can also be utilized for fault prediction, analysis of risk, calculation of results, testability, maintainability of software and study of consistency in the period of premature phases of software growth. These are also utilized in organizations for the purpose of risk reduction by forecasting the advantage of the software in the early stages of the SDLC (Software Development Lifecycle).

The defects occurring in software can be revealed by testers using several techniques. Major consequences that product defects cause are:- unnecessary time consumption and elevated cost [6], differences between actual and anticipated outcome [6, 7], creation of product not to the extent of customer's expectation level [8, 9], *etc*.. To avoid all these things and to achieve a good product, it is essential to forecast defects in the product and clear all faults. The scope of software fault forecasting is very huge and to explain it we need to describe the Software Defect Prediction (SDP). The SDP can be described as the practice of discovering those elements of any software system that are faulty [7, 10]. Software defect prediction models successfully forecasts the faults in software. These models utilize various software metrics to forecast defects accurately [11]. A good software defect prediction model is always useful for every software system. Initially, it augments the excellence of software & testing accuracy [6, 7] Secondly, it offers high satisfaction to the customers [6]. Thirdly, it aids in decreasing the price of correcting faults in final stages. [6] Finally, it supports in offering better software to consumers [9].

A. Soft Computing

For software defect prediction, several soft computing methods were recommended in the past [12]. Soft computing is a reference keyword to aggregate various computer science related mechanisms such as AI methods, Machine learning methods and various other mechanisms in the domain [13].

Soft Computing involves [13, 14]:

- Artificial Neural Network
- Neural Network
- Support Vector Machine
- Swam Intelligence : Ant Colony , Particle Swam Intelligence
- Machine Learning techniques[9]
- Probalistic Reasoning
- Decision Tree
- K-Nearest Neighbors
- Evolutionary Computation
- Evolutionary Algorithm : Genetic Algorithm
- Fuzzy Logic
- Bayesian Belief Network

Several statistical mechanisms such as Feature Subset Selection and PCA enhance the forecasting capacity in different SDP models [9]. Software metrics have vital role and it is supports the models to forecast the faults accurately. This review paper aids in providing the comprehensive information about the fault forecasting methods and about other related areas.

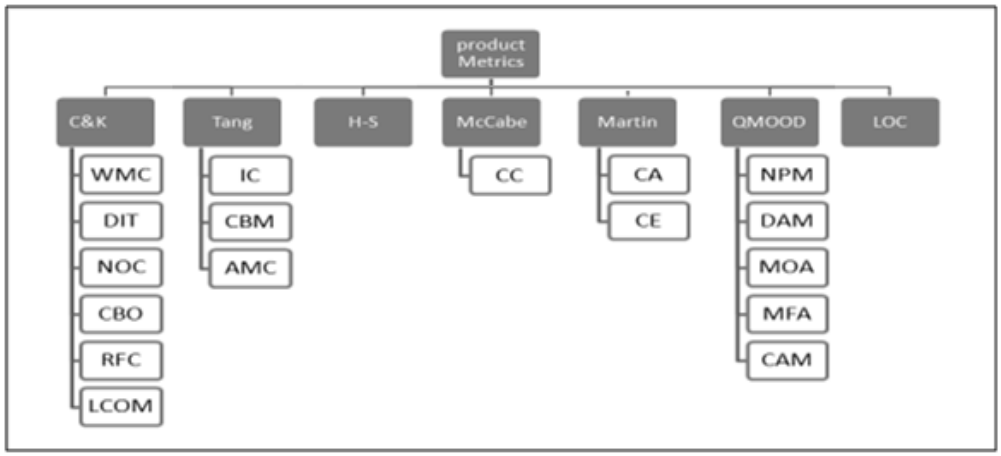

Figure 1. Characteristics of Different Metrics

B. Software Metrics

Software metric is a unit of measurement of an attribute in software or its specifications. These metrics are helpful in calculating the excellence of software. The software quality metrics [15] is a component of software metrics that concentrate on the quality features of the product, procedures & overall application. Various properties of products such are volume, architectural design, its computational complexity, efficiency and quality are described by product metrics. Process metrics are utilized by organizations to make software development better and in various support tasks like finding faults in product, bug fixing while development, fault discovery while testing and minimizing defect removal time. The project metrics defines the project features & implementation that involves the number of software developers, recruitment pattern in the life cycle of the software and, price, project plan and efficiency.

Various product metrics are [16]:

1. Chidamber and Kemerer [17].
2. The quality oriented extension to Chidamber & Kemerer metrics suite suggested by Tang *et al*. [18].
3. Cohesion in Methods (LCOM3) suggested by Henderson-Sellers [19].
4. On the basis of McCabe's complexity metric the class level metrics built [20].
5. Martin suggested Coupling metrics [21].
6. The Bansiya and Davis recommended QMOOD metrics suite [22].
7. Lines of Code (LOC)

C***. Software Defect Consequences***

In year 2008 and 2009, SANS institute carried out a research to spot the most frequent and hazardous twenty-five software defects or faults. Almost thirty companies cooperated for the research which includes software giants such as Oracle, Microsoft, Apple, Red Hat, CERT, Homeland Security, Breach Security, Tata, Aspect Security and MITRE; Educational establishments like Perdue University, University of California, *etc.* also participated in this research. These twenty-five security issues were categorized into three domains [23] demonstrated in Figure 2.

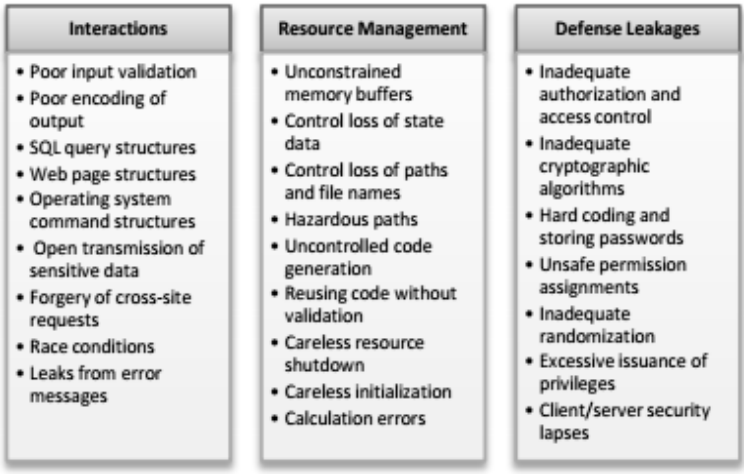

Figure 2. Defects Resultant Security Problems

Consequently it can be stated that defect forecasting is really vital in the area of reliability and quality of software. Defect forecasting is relatively a new survey in the area of software quality engineering. By wrapping up the key predictors and the collected data with the fault prediction model, the inter-dependencies between faults and predictor can be discovered. This paper provides extensive information about software defect prediction using various techniques and scope for future research.

D. Common Defect Prediction Procedure

It is a basic need for any forecasting system to have defect data and measurement data accumulated from real-time product testing to utilize as the learning dataset. There exist a ratio between a model's ability to discover defects in new dataset and well fitment of learning dataset with the model. Thus, assessment of a model's performance can be compared by analyzing forecasted faultiness of the modules in a test dataset against their real faultiness [24].Sunghun *et al*. [25] have illustrated a general defect forecasting procedure as shown in Figure 3.

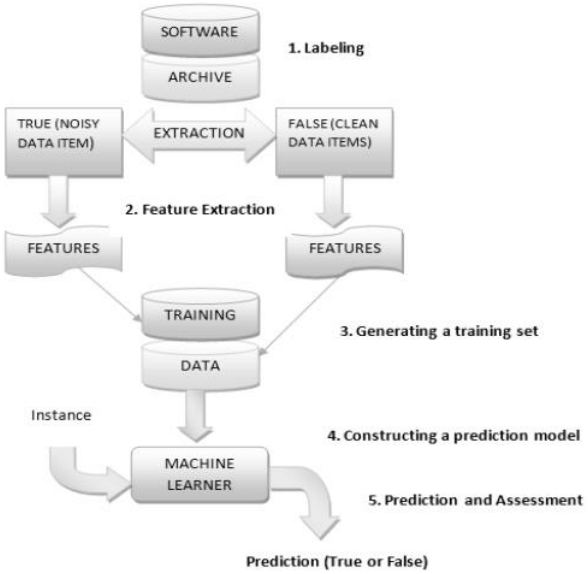

Figure 3. Common Defect Prediction Procedure

Labeling: To provide training to prediction model, faulty data has to be accumulated. To perform this step, typically we need to perform instance extraction (data items) from product archives and then Boolean labeling is done.

Feature Extraction & Training Set Construction: This phase includes feature extraction for forecasting labels of the instances. Common features for fault forecasting are structural dependencies, phrases, complexity metrics and modifications. Through integrating instance's features and labels, a training dataset can be produced that can be employed by any machine learning algorithm to build a defect forecasting model.

Constructing Forecasting Models: Common machine learning algorithms, like SVM (Support Vector Machines) or Bayesian Network trained on training dataset can easily be employed to construct a forecasting model. The model, subsequently, can work on new record to forecast its label as TRUE or FALSE.

Assessment: The assessment of a defect forecasting model requires a testing datas*et al*ong with a training dataset. The instance labels in the testing dataset are forecasted and the forecasting model is evaluated by comparing the prediction results and real labels. 10-fold cross-verification is generally employed to isolate the testing and training datasets.

2. Literature Survey

The empirical literature review of software fault forecasting models & methods are illustrated in this segment. With the intention to find future prospects in the aforementioned area of defect prediction, a comprehensive survey of various techniques introduced by different researchers is performed and brief of it is mentioned below.

Researchers EzgiErturk *et al*. [26] in 2014 offered a latest technique called Adaptive Neuron Fuzzy Inference System (ANFIS) for efficient software defect forecasting. For the testing purpose data was obtained through PROMISE Software Engineering Repository where McCabe metrics were utilized due to its characteristic of analyzing the programming effort thoroughly. The outcomes achieved were approx. 0.87, 0.78 and 0.86 for the ANN, SVM & ANFIS techniques, respectively.

Mie ThetThwin [27] in their model has applied 2 types of neural network methods. Primarily, it directs to forecast the volume of faults in class level and next it forecasts the number of lines modified per class. The General Regression neural network (GRNN) and Ward Neural Network (WNN) are the 2 types of neural network models that were utilized to work on the analysis of results based on the NASA dataset.

Jaweria Kanwal [28] has offered a classification based method to generate a priority oriented bug recommender system, which allots a priority level to the newly discovered bugs in the bug repository. It helps to find and resolve the bugs with their priority level. Mainly this technique utilizes machine learning related methods like SMV for the integral task of bug priority assignment for the new bug reports in the open source bug repository. It analyzes the bug report characteristics to predict the bug priority and analyzes the effect of training database size on the accurateness of bug priority recommender. Experimental results on this recommender model utilizing precision and recall measures demonstrated positive outcomes to get automatic bug priority allotment.

Xiao-dong Mu *et al*. [29] described in their model that, to enhance the accurateness of software defect forecasting, a co- evolutionary algorithm can be used which is based on the competitive construction. To operate with this algorithm, initially, competition method is used to construct co-evolutionary algorithm. In next phase, three evolution operators *i.e*. disturbed operators, allied operators and reduced operators are implemented for assessing the population, while, competition is used for computing the fitness in population. Evolutionary assessment has proved the efficiency of this model to enhance accuracy in software defect prediction.

Garcia *et al*. [30] designed forecasting models that uses decision trees to forecast whether if a bug is a blocking bug. To assess the models, researcher utilized fourteen

factors from the bug databases of six huge open source projects. Their assessment utilized decision trees to find which factors best detect these blocking bugs. The main aim of their research was to assist developers to discover these blocking bugs in shorter period of time. Their outcome illustrates that proposed forecasting models attain improved F-measures. They even discovered that most significant factors to catch blocking bugs are the size of comment and its text, the count of developers in the CC list of the bug report and the reporters knowledge. The discovered that, when compared to non-blocked bugs, the blocking bugs takes double to triple time to get fixed. Their research demonstrated that their model decreases the median time to discover a blocking bug.

Authors Askari *et al*. [31] utilized ANN *i.e*. Artificial Neural Network to enhance the generalization ability of the algorithm while forecasting the software faults. Subsequently, SVM *i.e*. Support Vector Machine mechanism was applied along with learning algorithms and evolutionary mechanisms. As a result it increased the classification margin and avoided over-fitting issues. From NASA datasets, eleven machine learning models were tested using this algorithm and results has demonstrated that it offers enhanced accuracy and precision compared to the other models.

Y.Chen *et al*. [32], based on the data mining methods & models designed a standard software fault management system. They made use of three data mining mechanisms *i.e*. classification, clustering; and association in combination with two popular data mining models *i.e*. Bayesian Network and Probabilistic Relational Model in the proposed methodology of their work.

Mrs. Agasta Adline *et al*. [33] in 2014 demonstrated that when defect labels for modules are unavailable, the forecasting of defect proneness of program modules are difficult job and this issue is progressively increasing in the IT companies. They tried to forecast the defect proneness of a program module when the defect labels of modules do not exist. For classification, the supervised algorithms such as Genetic algorithm were used to forecast software defects.

Kaur *et al*. [34] discovered various data mining methods that were helpful for software engineering works in programming, evaluating, bug discovery, debugging & support. In their review they presented the concepts in data mining and particularly about clustering methodologies. They state that each method has ability to solve various issues and has its own merits and demerits. As per their research presently there are no single clustering technique or algorithm which are able to solve the all the issues and is suitable for all kinds of software.

Ahmet Okutan *et al*. [35] presented a latest kernel system to forecast the volume of faults in the software modules such as in classes or in files. The proposed system was based on a pre-calculated kernel matrix that works on the resemblance between the modules of the software system. New kernel techniques with primitive kernels (linear & RBF kernels) in the survey were compared and demonstrated that it attains competitive outcomes. Moreover, the presented fault forecasting system in their model is also comparable to popular fault forecasting models such as, IBK & linear regression. It was observed that before test phase or maintenance, developers can apply presented technique to effortlessly forecast the faultiest modules in the software system and concentrate on them before testing all the modules in the software. This technique can reduce the testing task as well as the complete cost of the project automatically.

Yajnaseni Dash *et al*. [36] researched various mechanisms for the forecasting of OO metrics through utilizing neural networks. In case of object oriented metrics, this methods was surveyed to be optimal appropriate for forecasting. Compared to other artificial intelligence methods, the neural network utilized less estimation task and it obtains enhanced demonstration ability as well as competency of work in difficult functions also.

Ms. Pallavi and Ms. Puneet Jai Kaur in 2013 [37] utilized different kinds of data mining methods such as association mining, classification and clustering mining to forecast the software faults and this assisted the software engineers in developing

enhanced models. The investigation revealed that Unsupervised methods can be applied for model development when fault labels are unavailable.

Researcher Xiaoxing Yang, *et al* in 2014 [38] applied rank performance optimization method for software prediction model development. In this paper, rank to learning method was utilized. The model was developed on basis of earlier model and later it was investigated for the purpose of enhancing the performance of the model. The task consists of 2 facets *i.e.*, an innovative application of the learning-to-rank methodology to real-world datasets for fault forecasting and another is thorough assessment and comparison of the learning-to-rank approach against other techniques that are utilized for forecasting the allocation of modules as per the forecasted count of faults in them. Their investigation revealed the effect of optimization on model efficiency by applying rank-to-learning methodology that actually enhances the forecasting precision.

3. Constraints & Limitations

Defect prediction models perform better when volume of training dataset present in software repository is very good. A suitable data mining model can only performs well in forecasting the defects in forecasting models. But still there is a dreadful requirement of best data mining model to predict the faults from the software bug repository perfectly.

a) Extremely skewed & unbalanced datasets

Current prediction models based on un-sampling and training databases doesn't hold any information about no. of defects, fault distributions in each module and segregation of defects among modules. [39]

- There is a shortage of business knowledge in data mining algorithms and causes serious performance issues when it is unable to retrieve required information concerned to software metrics with defect frequencies [40].
- Generally low performance by fault forecasting models are due to imbalance in training datasets [41].
- Another important reason for under-performance by defect forecasting models is extremely skewed dataset. Though, the results of highly balanced dataset are also not so satisfactory. [42].

b) Early life cycle and multiple dataset

- For recognition of defect prone modules, the early life cycle data is not helpful. [43,44]
- When various software repositories are mined, there is no chance for modification in outcome of fault forecasting.[45]
- The solitary classifier is systematically not suitable to make utilization of each and every feature, so the merging of various classifier still remain uncertain [46].

c) Huge counts of features & high level of software modules

- Removal of faults from the dataset is not feasible by most of the machine leading algorithms that stores continuous features [47].
- For fault forecasting, supervised algorithms are more helpful at similar logical level but it is not fit for elevated level software modules [48].
- In practical application, present classifier based fault forecasting model are somewhat inaccurate as it considers huge count of features [49].

d) Accurate fault forecasting model

- For huge scale software system, an accurate fault forecasting model is mandatory that is more robust to noise [50].
- For classification of faulty & non-faulty modules, the conventional decision trees are employed. Though, the conventional decision tree system poses various demerits [51].
- For the purpose of forecasting software faults from the bug repository, the proper data mining model is required [52]

- The information transformation can enhance the efficiency of software quality models [53].

e) Reliable data mining mechanism

- Currently, an open problem is unavailability of an ideal data mining method to construct best forecasting model [54].
- Testers and QA experts are unable to find suitable fault prediction methods due to lack of good comparative analysis on popular techniques. [55].
- Assessment testing of various forecasting model is still a universal problem and effort reduction gain utilizing the models is overlooked while its assessment [56].
- Diverse defect forecasting methods are proposed but the consistency in accuracy is not proven [57].
- A proper data mining method to create an enhanced forecasting model is an open challenge [58].
- To evaluate the quality of software, various forecasting methods were utilized, but still we have a shortage of comparative research to estimate the accuracy of different model [32].

4. Comparative Analysis

This segment demonstrates the comparative study of different methods utilized for software fault forecasting with their merits & demerits. Below mentioned Table 1 clearly explains the comparative research outcome

Table 1. Comparative Analysis

Methods	Dataset Utilized	Merits	Demerits
ANN	NASA AR1.AR6 AND MDP	Needless to make out metrics association. It has self learning ability, hence it has better accuracy	It can't manage imprecise information
Clustering	NASA MDP Repository	It is appropriate for small databases	Database should be unlabeled
Association Rule Mining	NASA MDP Repository	Rules creation utilizing previous data and forecast the defects	Needs continuous value of software metrics
Decision Tree	NASA AR1,AR6	Running operations on tree structure, thus, gives better accuracy in outcomes compared to others	Structuring of decision tree is complex
SVM	NASAAR1. AR6	When Utilizing diverse kernel functions it offers good forecasting outcomes	Not appropriate for huge no. of software metrics

5. Conclusion

In this survey, we reviewed various defect prediction algorithms based on the various machine learning classifiers and diverse feature selection methods. We studied the benefits & limitations of Association Rule Mining, Support Vector Machine, Decision Tree, Artificial Neural Network and Clustering Machine Learning Methods for forecasting software faults and also found that defect prediction models perform better when the volume of training dataset present in software repository is very good. Ranked with the ability and performance, the different methods utilized for fault forecasting in software systems can be organized as, best in class is Neural Network, then Decision Tree, Bayesian Network and subsequent to it is Support Vector Machine. The Apriori Algorithm and Genetic Algorithm for fault forecasting methods are not studied much. It is purely recommended that, the Ensemble Machine learning and one class SVM are the areas that can be utilized widely in future but as per this research, SVM method to some extent doesn't work well, therefore Ensemble Learning can be enhanced in future to forecast faults.

References

[1] Tian, J., "Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement" John Wiley & Sons,(2005).
[2] Xing, F. , Guo, P. , Lyu, M. R. "A Novel Method for Early Software Quality Prediction Based on Support Vector Machine". 2005,In: Proceedings of The 16th IEEE International Symposium on Software Reliability Engineering
[3] Emam, K.,El., "The ROI from Software Quality". Auerbach Publications, Taylor and Francis Group, LLC, (2005).
[4] Khoshgoftaar, T.M., Allen, E.B., Kalaichelvan, K.S., Goel, N., "Early Quality Prediction: A Case Study in Telecommunications".2006, IEEE Software.
[5] http://www.softwaretestingtimes.com /2010/04/ software testingeffort-estimation.htm
[6] K. Punitha, Dr. S. Chitra, "Software Defect Prediction Using Software Metrics-A Survey," IEEE International Conference on Information Communication and Embedded Systems, pp. 555-558, 2013
[7] P.A. Selvaraj, Dr. P. Thangaraj, "Support Vector Machine for Software Defect Prediction," International Journal of Engineering & Technology Research, Vol. 1, Issue 2, pp. 68-76, 2013
[8] P. Paramshetti, D.A. Phalke, "Survey On Software Defect Prediction Using Machine Learning Techniques," International Journal Of Science And Research, Vol. 3, No. 12, pp. 1394-1397, 2014
[9] M. Gayathri, A. Sudha, "Software Defect Prediction System Using Multilayer Perceptron Neural Network With Data Mining," International Journal Of Recent Technology And Engineering, Vol. 3, No. 2, pp 54-59, 2014
[10] Er. R. Mahajan, Dr. S.K. Gupta, R.K. Bedi, "Comparison Of Various Approaches Of Software Fault Prediction: A Review," International Journal Of Advanced Technology & Engineering Research , Vol. 4 , No. 4, pp. 13-16, 2014.
[11] Y.Xia, G.Yan, Q.Si, "A Study on the Significance Of Software Metrics In Defect Prediction," IEEE Sixth International Symposium On Computational Intelligence And Design, pp. 343-346, 2013.
[12] N. Gayathri, S. Nikolas, A.V. Reddy, "Feature Selection Using Decision Tree Induction in Class Level Metrics Dataset for Software Defect Predictions," In Proceedings of the World Congress on Engineering and Computer Science, Vol. 1, 2010
[13] M.D. Ambros, M. Lanza, R. Robbes, "An Extensive Comparison of Bug Prediction Approaches," IEEE Seventh International Conference on Mining Software Repositories, pp. 31-41, 2010.
[14] K. Sankar, S. Kannan, P. Jennifer, "Prediction Of Code Fault Using Naïve Bayes And SVM Classifiers," Middle-East Journal Of Scientific Research, Vol. 20, No. 1, pp. 108-113, 2014.
[15] www.pearsonhighered.com/samplechapter4/softw-are quality metrics overview
[16] L. Madeyski, M. Jureczko, "Which process metrics can significantly improve defect prediction models? An empirical study," (2014)
[17] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design," in IEEE Transactions on Software Engineering, vol. 20, no. 6, pp. 476-493, Jun 1994.
[18] Tang MH, Kao MH, Chen MH (1999) An Empirical Study on Object-Oriented Metrics. In: METRICS '99: Proceedings of the 6th International Symposium on Software Metrics, IEEE Computer Society, Washington, DC, USA, p 242
[19] B. Henderson-Sellers and I. Graham, "OPEN: toward method convergence?," in Computer, vol. 29, no. 4, pp. 86-89, Apr 1996.
[20] McCabe T (1976) A Complexity Measure. IEEE Transactions on Software Engineering 2:308–320

[21] Martin R (1994) OO Design Quality Metrics - An Analysis of Dependencies. In: OOPSLA'94: Proceedings of Workshop Pragmatic and Theoretical Directions in Object-Oriented Software Metrics, pp 1–8.
[22] Bansiya J. and C. G. Davis (2002): A Hierarchical Model for Object-Oriented Design Quality Assessment, IEEE Transactions on Software Engineering, pp. 4-17, 2002
[23] Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies, the McGraw-Hill Companies, 2010, ISBN: 9780071621618.
[24] A. Güneş Koru and Hongfang Liu ,"Building Effective Defect-Prediction Models in Practice" IEEE, Vol. 22, No. 6 November/December 2005.
[25] Sunghun Kim, Hongyu Zhang, Rongxin Wu and Liang Gong, "Dealing with Noise in Defect Prediction", CSE"11, Waikiki, Honolulu, HI, USA, ACM 978-1-4503-0445-0/11/05, 2011
[26] E. Erturk and E. A. Sezer, "A comparison of some soft computing methods for software fault prediction," Expert Systems with Applications, 2014
[27] M. M. T. Thwin and T.S. Quah, "Application of neural networks for software quality prediction using object-oriented metrics," Journal systems and software, vol. 76, no. 2, pp. 147-156, 2005
[28] Jaweria Kanwal OM (2010), "Managing open bug repositories through bug report prioritization using SVMs", Proceedings of the 4th international conference on open-source systems and technologies, ICOSST, pp 1–7
[29] Xiao-dong Mu, Rui-hua Chang, Li Zhang, "Software Defect Prediction Based on Competitive Organization Co-Evolutionary Algorithm", Journal of Convergence Information Technology(JCIT) Volume7, Number5, (2012).
[30] Garcia HV, Shihab E (2014) Characterizing and predicting blocking bugs in open source projects categories and subject descriptors. In: Proceedings of the 11th working conference on mining software repositories,pp 72–81
[31] 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.
[32] Y.Chen, X.Shen, P.Du and B.Ge, "Research on software defect prediction based on data mining", The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, pp.563-567.
[33] 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.
[34] Kaur M, Garg SK (2014) Survey on clustering techniques in data mining for software engineering. International Journal of Advanced and Innovative Research, Vol. 3, Issue 4, pp.238–243.
[35] Ahmet Okutan1 and Olcay Taner Yıldız, (2013), "A Novel Regression Method for Software Defect Prediction with Kernel Methods", ICPMRA 2013 - International Conference on Pattern Recognition Applications and Methods, pp 216-221.
[36] 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.
[37] 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.
[38] Xiaoxing Yang, et.al. (2014), IEEE TRANSACTIONS ON RELIABILITY, This article has been accepted for inclusion in a future issue of this journal.
[39] L. Li and H. Leung, "Using the Number of Faults to Improve Fault-Proneness Prediction of the Probability Models", Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering - Volume 07, pp.722-726.[40] T. Menzies, B. Turhan, A. Bener, G. Gay, B. Cukic and Y. Jiang, "Implications of ceiling effects in defect predictors", Proceedings of the 4th international workshop on Predictor models in software engineering, 2008,Leipzig,Germany , pp.47.
[40] Y. Kamei, A. Monden, S. Matsumoto, T. Kakimoto and K. Matsumoto, "The effect of over and under sampling on fault–prone module detection", First International Symposium on Empirical Software Engineering and Measurement, 2007, ESEM 2007, pp.196-204
[41] Z. Li and M. Reformat, "A practical method for the software fault-prediction", IEEE International Conference on Information Reuse and Integration, 2007. IRI 2007, pp.659-666
[42] P.S. Sandhu, R. Goel, A.S. Brar, J. Kaur and S. Anand, "A Model for Early Prediction of Faults in Software Systems", The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010 , pp.281-285
[43] Y. Jiang , B. Cukic and T. Menzies, "Fault Prediction using Early Lifecycle Data", Proceedings of The 18th IEEE International Symposium on Software Reliability 2007, pp.237-246.
[44] R. Ramler, S. Larndorfer and T. Natschläger, "What Software Repositories Should Be Mined for Defect Predictors?", Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications , pp.181-187.
[45] T.M. Khoshgoftaar, P. Rebours and N. Seliya, "Software quality analysis by combining multiple projects and learners", Software quality journal 2009, Springer, vol. 17, pp.25-49

[46] P. Singh and S. Verma, "An Investigation of the Effect of Discretization on Defect Prediction Using Static Measures", IEEE International Conference on Advances in Computing, Control, and Telecommunication Technologies (2009), pp.837- 839
[47] P. Huang and J. Zhu, "Predicting Defect-Prone Software Modules at Different Logical Levels", International Conference on Research Challenges in Computer Science,2009.ICRCCS '09, pp.37 - 40
[48] S. Shivaji, E.J. Whitehead, R. Akella and S. Kim, "Reducing Features to Improve Bug Prediction", 24th IEEE/ACM International Conference on Automated Software Engineering, ASE'09, pp.600-604.
[49] L. Guo, Y. Ma, B. Cukic and H. Singh, "Robust prediction of fault-proneness by random forests", Proceedings of the 15th International Symposium on Software Reliability Engineering,2004, pp.417-428.
[50] M. Chis, "Evolutionary Decision Trees and Software Metrics for Module Defects Identification", International Journal of Computer and Information Science,2008, pp. 273-277.
[51] N. K. Nagwani and S. Verma ,"Predictive Data Mining Model for Software Bug Estimation Using Average Weighted Similarity" , 2010 IEEE 2nd International Advance Computing Conference, pp.373-378.
[52] Y. Jiang, B. Cukic and T. Menzies, "Can data transformation help in the detection of fault-prone modules?", Proceedings of the 2008 workshop on Defects in large software systems, July 20-20, 2008, Seattle, Washington ,pp.16-20.
[53] E. Arisholm, L.C. Briand and M. Fuglerud, "Data Mining Techniques for Building Fault-proneness Models in Telecom Java Software", Proceedings of The 18th IEEE International Symposium on Software Reliability, pp.215-224.
[54] N. Gayatri, S. Nickolas, A.V. Reddy and R. Chitra, "Performance Analysis of Data Mining Algorithms for Software Quality Prediction", International Conference on Advances in Recent Technologies in Communication and Computing, 2009, ART Com '09, pp.393 – 395.
[55] T. Mende and R. Koschke, "Revisiting the Evaluation of Defect Prediction Models", Proceedings of the 5th International Conference on Predictor Models in Software Engineering 2009.
[56] V.U.B. Challagulla, F.B. Bastani, I. Yen and R.A. paul, "Empirical Assessment of Machine Learning based Software Defect Prediction Techniques", Proceedings of the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, 2005, pp.263-270.
[57] X. Zhao, Y. Liu and S. Yong, "Predicting Software Defects using Multiple Criteria Linear Programming", Proceedings of the International Symposium on Intelligent Information Systems and Applications (IISA'09)2009, pp.583-585

Authors

R. Jayanthi, she is working as an Assistant Professor at PESIT, BSC, Bangalore, India. She is pursuing her PhD from Bharathiyar University. Her research area is Software Engineering and Neural Networks

Lilly Florence, she is working as a Professor in MCA department, at Adiyamman College of Engineering, Tamilnadu, India. She has completed her PhD in Computer Science from Mother Theresa University, Tamilnadu. She has more than 20 publications in reputed conferences and journals. Her area of interest includes Artificial Intelligence, Software Reliability and Data Mining *etc.*

Arti Arya, she is working as a Professor and Head of the Department of MCA, at PESIT, BSC, Bangalore, India. She has Bachelor's and Master's in Mathematics from Delhi University. She has completed her PhD in Computer Science Engineering from MDU in 2009. She has more than 20 publications in reputed Conferences and Journals. Her area of interest includes Data Mining, Text Mining, Artificial Intelligence, knowledge Management, Big Data Analytics *etc.*