

Predicting Defect of Software System

Soumi Ghosh, Ajay Rana and Vineet Kansal

Abstract Any particular study on software quality with all desirable attributes of software products can be treated as complete and perfect provided it is defective. Defects continue to be an emerging problem that leads to failure and unexpected behaviour of the system. Prediction of defect in software system in the initial stage may be favourable to a great extent in the process of finding out defects and making the software system efficient, defect-free and improving its over-all quality. To analyze and compare the work done by the researchers on predicting defects of software system, it is necessary to have a look on their varied work. The most frequently used methodologies for predicting defects in the software system have been highlighted in this paper and it has been observed that use of public datasets were considerably more than use of private datasets. On the basis of over-all findings, the key analysis and challenging issues have been identified which will help and encourage further work in this field with application of newer and more effective methodologies.

Keywords Software defects • Defect prediction • Software datasets • Data mining • Machine learning

S. Ghosh (✉) • A. Rana
Amity University Uttar Pradesh, Sec-125, Noida, India
e-mail: soumighoshphd@gmail.com

A. Rana
e-mail: ajay_rana@amity.edu

V. Kansal
I. T. S. Engineering College, Knowledge Park III, Greater Noida, India
e-mail: dir.engg@its.edu.in

1 Introduction

In the arena of software engineering, a crucial problem like prediction of defects is often taken into account as a very important step for the purpose of quality improvement obtained in lesser period and by minimum cost. Prediction of defects is highly necessary in order to find out sensitive and defect-prone domains in the stage of software testing, so that it may help in qualitative improvement of the software system with reduced cost. The possibility of detection of potential faults in software system at an early stage may help in effective planning, controlling and execution of software development activities to a considerable extent. In modern days, as the development of software has become very meaningful and keeping pace with the necessity, it may be safely said that hence reviewing and testing of the software system will be very essential and result-oriented in the case of predicting defects. Predicting software defects often involves huge cost and as-such the matter of correction of software defects is altogether a very expensive matter [25]. Those studies which have been carried out in the recent years, reveal the fact that the case of predicting defects assumes more importance compare to testing and reviewing process of software systems [36, 43]. As such, accuracy in predicting software defects is certainly very much helpful in case of improving software testing, minimizing the expenses [10] and improving the software quality [22].

In this paper, although analysis and comparison of various research work (from the year 1992 to 2015) on predicting software defects by using various methodologies have been made but only those unique and most updated methodologies (year 2005–2015) have been highlighted. This paper is having the objective of critically estimate the efficacy of the methods adopted in predicting software defects. Simultaneously, evaluation of the varied systems in prediction of software defects have been made and thus realized the effectiveness and importance of some methodologies like Advance Machine Learning, Neural Network and Support Vector Machine applied most frequently compared to various other techniques for achieving desirable accuracy in predicting defects in the software system. This paper has also highlighted the requirement of further work in this field by applying newer methodologies since the previous ones have not at all been found defect-free or at-least a least defective software system which may finally produce quality software system.

2 Literature Review

In order to perform the analysis, we explored 102 papers (during the period 1992–2015) from various digital library like IEEE Transactions on Software Engineering, ACM, Springer, Elsevier, Science Direct, International Conferences, Reports, Thesis and even technical papers and case studies were also reviewed. After exploring these digital libraries, we found that most of the research work on predicting defects of software system was performed on similar patterns/methodologies/

techniques as well as on nearly same datasets. As such, papers based on similar patterns/methodologies/techniques, datasets were excluded. We included only 49 those papers which are found unique and updated (from the year 2005 to 2015) in this particular field. Since 1992 various methodologies have been applied in predicting defects of software system. But in modern days, various methodologies are basically very favourable in predicting defects in software system. Only those methodologies which were considered unique as well as updated, have been analyzed, compared and the results obtained would help to determine which are the most frequently used and effective methodologies in the field of predicting defects of software system.

2.1 Predicting Defects of Software System Using Data Mining (DM)

Campan et al. [9] experimented with Length Ordinal Association Rule in datasets for searching out any interesting new rules. Song et al. [44] emphasized on Rule Mining methodologies in predicting and correcting software defects. Kamei et al. [26] proposed a methodology combining Logistic Regression analysis with Association Rule Mining for predicting software defects. Chang et al. [12] combined Decision Tree and Classification methodologies-Action Based Defect Prediction (ABDP) along with Association Rule Mining for predicting and discovering software defects pattern with minimum support and confidence. Gray et al. [21] experimented with Support Vector Machine (SVM) classifier based on Static Code Metrics and NASA datasets to maintain defective classes and remove redundant instances. Riquelme et al. [39] applied Genetic Algorithms finding rules featuring subgroups predicting defects and extracted software metrics program dataset from the Promise repository. Gayatri et al. [19] combined Induction methodology with Decision Tree and the new method of feature selection was better as compared to SVM and RELIEF methodologies. Gray et al. [20] analyzed Support Vector Machine (SVM) classifiers based on NASA datasets in such a way that identifies software defects and the basic idea was to classify training data rather than obtaining test datasets. Liu et al. [34] experimented with a new Genetic Programming based search methodology for evaluating the quality of software systems. It found that Validation cum Voting classifier was better than Baseline classifier, Validation classifier. Tao and Wei-Hua [46] found that Multi-Variants GAUSS Naive Bayes methodology was superior as compared to other versions of Naive Bayes methods and J48 algorithm in predicting defects of software system. Catal [11] reviewed different methodologies such as Logistic Regression, Classification Trees, Optimised Set Reduction (OSR), Artificial Neural Networks and discriminate model used during the period 1990 to 2009 on predicting software defects. Kaur and Sandhu [28] found that accuracy level was on higher side in case of software system based on K-Means. Tan et al. [45] attempted prediction of software defects by application of functional cluster of programs vide class or file which significantly improved recall and precision percentage.

Dhiman et al. [15] used a clustered approach in which the software defects will be categorized and measured separately in each cluster. Kaur and Kumar [30] applied clustering methodology for forecasting as well as error forecasting in object-oriented software systems. Najadat and Alsmadi [37] proved Ridor algorithm with other classification approaches on NASA datasets to be an effective methodology for predicting software defects with higher accuracy level. Sehgal et al. [41] focused on application of J48 algorithm of Decision Tree methodology in prediction of defects in software systems. The performance of new methodology was evaluated against the IDE algorithm as well as Natural Growing Gas (NGG) methodology. Banga [7] found that a hybrid architecture methodology called as GP-GMDH or GMDH-GP was more effective as compared to other methodologies on the ISBSG datasets. Chug and Dhall [14] different methodologies were used on different datasets of NASA with both supervised and unsupervised learning methodologies for defect prediction. Okutan and Yildiz [38] for predicting software defects proposed a kernel methodology based on pre-computed kernel metrics. It was observed that the proposed defect prediction methodology was also comparable with other existing methodologies like Linear Regression and IBK. Selvaraj and Thangaraj [42] predicted software defects using SVM and compared its effectiveness with Naive Bayes and Decisions stumps methodologies. Adline and Ramachandran [3] proposed program modules for predicting the fault-proneness when the fault levels of modules are not available. The supervised methodologies like Genetic Algorithm for classification and predicting fault in software were applied. Agarwal and Tomar [4] observed that Linear Twin Support Vector Machine (LTSVM) on the basis of feature selection and F-score methodology was superior to other methodologies. Sankar et al. [40] advocated feature selection methodology using SVM and Naive Baye classifier based on F-mean metrics for predicting and measuring the defects in software system.

2.2 Predicting Defects of Software System Using Machine Learning (ML)

Boetticher [8] analyzed K-Nearest Neighbour (K-NN) algorithm or sampling for predicting software defects and its performance was not effective in case of small datasets. Ardil et al. [5] applied one of the easiest forms of Artificial Neural Network and compared it with other modules of Neural Network. Chen et al. [13] predicted software defects using Bayesian Network and Probabilistic Relational Models (PRM). Jianhong et al. [23] showed that the Resilient Back propagation algorithm based on neural network was superior methodology for predicting software defects. Xu et al. [47] evaluated the effectiveness of software metrics in predicting software defects by applying various Statistical and Machine Learning methodologies. Gao and Khoshgoftaar [17] predicted software defects by use of class-imbalanced and high dimensional database system. In this approach, modelling and feature selection was done on the basis of alternative use of both original and sampled data. Li et al. [32] found that effectiveness of sampled based methodologies like active

semi-supervised methodology called as ACoForest was better compared to Random Sampling both with conventional machine learners and semi-supervised learner. Kaur [29] used software metrics along with Neural Network to find out those modules suitable for multiple uses. Abaei and Selamat [1] experimented with the application of various machine learning and artificial intelligent methodologies on different public NASA datasets in connection with predicting software defects. Askari and Bardsiri [6] predicted software defects by using Multilayer Neural Network. Support Vector Machine with the Learning algorithm and Evolutionary methodologies were also used for the purpose of removing the defects. Gayathri and Sudha [18] applied Bell function based Multi-Layer Perceptron Neural Network along with Data Mining for predicting defects in software system and its performance was compared with other Machine Learning methodologies. Jing et al. [24] proposed an efficient model using Advanced Machine Learning methodology- Collaborative representation classification for Software Defect Prediction (CSDP). Kaur and Kaur [27] predicted defects in classes using Machine Learning methodologies with different classifiers. Li and Wang [33] compared various Ensemble Learning methodologies- Ada Boost and Smooth Boost with SVM, KNN, Naive Baye, Logistic and C4.5 for predicting software fault proneness on imbalanced NASA data sets. Malhotra [35] predicted defects and estimated relationship among static code measures, different ML methodologies were applied. Yang et al. [48] used a Learning-to-Rank methodology for predicting defects in software system and also compared its effectiveness with others. Abaei et al. [2] studied the effectiveness of new version of semi-supervised methodology on eight datasets from NASA and Turkish in predicting software defects with high accuracy. Erturk and Sezer [16] proposed a new methodology-Adaptive Neuron Fuzzy Inference System (ANFIS) and compared it with other methodologies (SVM, ANN, ANFIS) using Promise repository for predicting software defects. Laradji et al. [31] Average Probability Ensemble (APE) comprised of seven classifiers was superior to weighted SVM and Random Forest methodologies. Finally, a new version of APE comprised of greedy forward selection was more efficient in removing duplicate and unnecessary features. Zhang et al. [49] predicted software efforts by using methodology based on Bayesian Regression Expectation Maximize (BREM).

3 Methodology

In this paper, a specific methodology was used with the aim of analyzing and comparing only those different, unique and updated methodologies (from the year 2005 to 2015) for predicting defects of software system. Different methodologies were compared on the basis of studies and the results showed that Advance Machine Learning, Neural Network and Support Vector Machine methodologies are the most commonly used techniques for predicting software defects. Summary of major findings are given in Table 1.

Table 1 Summary of major findings of different software defect prediction methodologies

S. no.	Authors	Methodologies	Findings
1.	Boetticher [8]	K-Nearest Neighbour	Its performance was not found satisfactory
2.	Song et al. [44]	Rule Mining, Naive Bayes, PART, C4.5	Accuracy level of proposed method was higher by at least 23 % than other methods
3.	Chang et al. [12]	Association Rule Mining with Action Based Defect Prediction	It was applicable to discover defects as well as to handle continuous attributes of actions
4.	Gray et al. [21]	Support Vector Machine Classifier and Static Code Metrics	It is highly effective and having an accuracy level of about 70 %
5.	Ardil et al. [5]	Artificial Neural Network	Defects were mostly found in the Neural Network modules
6.	Riquelme et al. [39]	Genetic Algorithm (GA) with Promise repository	GA is able to efficiently handle the unbalance datasets that consist of more non-defective than defective sample
7.	Gray et al. [20]	Support Vector Machine	Effectiveness of classifiers in separating training data was experimented
8.	Xu et al. [47]	Neuro-Fuzzy with ISBSG repository	Adapted methodology to improve the accuracy of estimation
9.	Gayatri et al. [19]	Decision Tree Induction method	Its performance is better than all other methodologies
10.	Tao and Wei-Hua [46]	Multi-Variants GAUSS	This method was better than all other Naive Bayes method
11.	Chen et al. [13]	Bayesian Network, Probabilistic Relational Models	Altogether a new model was proposed
12.	Liu et al. [34]	Genetic algorithm-Baseline, Validation, Validation cum Voting classifier	Proposed approach was more effective and accurate when applied with multiple datasets
13.	Jianhong et al. [23]	Resilient Back Propagation Algorithm	Proposed algorithm was applicable to identify the modules having major defects
14.	Catal [11]	Logistic Regression, Classification Trees, Optimised Set Reduction (OSR), Artificial Neural Network and Discriminate model	Performance of OSR methodology was found to be the best among all other methods with an accurate rate of 90 %
15.	Kaur and Sandhu [28]	K-Means clustering approach	It was found effective and having 62.4 % accuracy level in predicting software defects
16.	Gao and Khoshgoftaar [17]	Promise repository based on different criterions	A comparative estimate on effectiveness of all the criterions was made
17.	Li et al. [32]	Semi-supervised model-ACoForest	Defect prediction is not at all effected by size of the datasets

(continued)

Table 1 (continued)

S. no.	Authors	Methodologies	Findings
18.	Tan et al. [45]	Functional cluster approach	Applied for significantly improving recall from 31.6 to 99.2 % and precision from 73.8 to 91.6 %
19.	Najadat and Alsmadi [37]	Ridor algorithm	Better methodology than others and having more accuracy rate
20.	Kaur [29]	Neural Network	Model is effective in improving accuracy level
21.	Sehgal et al. [41]	Decision Tree, J48 algorithm	J48 algorithm is 93.32 % accurate than IDE algorithm as well as Natural Growing Gas (NGG) methodology had an accuracy level of 80 %
22.	Dhiman et al. [15]	Clustered approach	Effective in analyzing defects and integrating software modules
23.	Kaur and Kumar [30]	Clustering methodology	Forecasting error of software system
24.	Chug and Dhall [14]	Classification methods	Different parameters were followed for estimating the effectiveness
25.	Okutan and Yildiz [38]	Linear and RBF kernels	Effective model in case of reducing testing effort and also total project cost
26.	Banga [7]	GP-GMDH	GP-GMDH was superior to all other methodologies
27.	Selvaraj and Thangaraj [42]	SVM	SVM had better performance than Naive Bayes and Decisions stumps methodologies
28.	Adline and Ramachandran [3]	Genetic Algorithm	Software defect prediction under condition of unknown fault level of software
29.	Kaur and Kaur [27]	Machine Learning based on Classifier	J48 and Bagging methodologies were considered most effective than others
30.	Sankar et al. [40]	Support Vector Machine (SVM), Naive Bayes	Naive Bayes was found better than SVM
31.	Abaei and Selamat [1]	Machine Learning, Artificial Immune Systems (AISs)	For, large datasets-Random Forest showed better result. Small datasets-Naive Bayes was more effective. With Feature Selection-Immuno99 outperforms. In absence of Feature Selection-AIRS Parallel was effective
32.	Gayathri and Sudha [18]	Bell function based Multi-Layer Perceptron Neural Network	Proposed approach was effective with an accuracy level of 98.2 %

(continued)

Table 1 (continued)

S. no.	Authors	Methodologies	Findings
33.	Askari and Bardsiri [6]	Multilayer Neural Network	More accurate and précised compared to others
34.	Li and Wang [33]	SVM, KNN, Naive Baye, Logistic, C4.5, Ada Boost and Smooth Boost methods	Smooth Boost method was considered best and most effective as compared to other method
35.	Malhotra [35]	Machine Learning (ML) and Statistical logistic Regression (SLR)	ML methodologies were considered effective and better result oriented than LR methods
36.	Agarwal and Tomar [4]	Linear Twin Support Vector Machine (LTSVM)-F-score	LTSVM performed better than others
37.	Jing et al. [24]	Collaborative Representation Classification for Software Defect Prediction (CSDP)	Experimented with a new methodology for effective defect prediction with low cost
38.	Erturk and Sezer [16]	Adaptive Neuron Fuzzy Inference System (ANFIS), SVM, ANN	Comparative analysis of soft-computing methods found SVM, ANN, ANFIS having 0.7795, 0.8685 and 0.8573 respectively
39.	Abaei et al. [2]	Semi-supervised methodology	Considered as helpful and effective automated methodology for detecting defects
40.	Laradji et al. [31]	Ensemble Learning Method on selected features-Average Probability Ensemble (APE) model with different classifiers	APE model showed better result than SVM, Random Forest method

Figure 1 indicates different methodologies used in software defect prediction from the year 2005 to 2015. This illustrates that these methodologies have been compared on the basis of studies and the results showed that Advance Machine Learning, Neural Network and Support Vector Machine techniques are the most frequently used as compared to other techniques in predicting defects of software system.

The Fig. 2 shows the datasets used in software defect prediction. The research studies using public datasets comprise 64.79 % whereas studies using private datasets cover 35.21 %. In-fact, the public free distributed datasets are mostly connected with PROMISE Repository and NASA Metrics Data Program. Private Datasets are not distributed as public datasets and they basically belong to private companies.

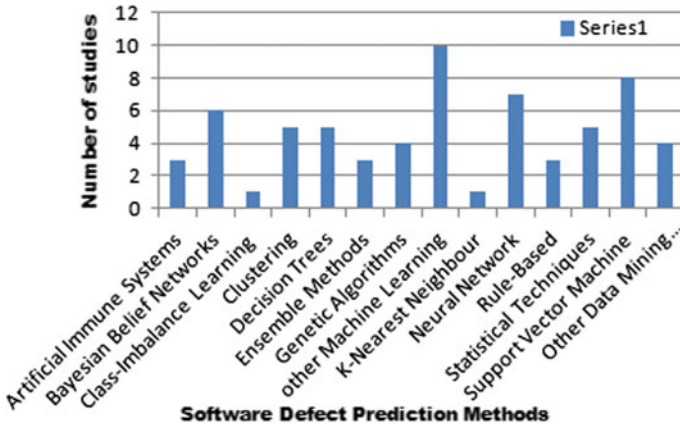
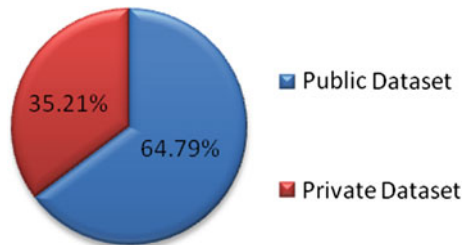


Fig. 1 Methods used in software defect prediction

Fig. 2 Datasets used in software defect prediction



4 Key Analysis

The analysis of various techniques applied for software defect prediction till date has brought out the following observations:-

- (a) Proper prediction of software defects in the initial phase of design level of software development lifecycle can improve software quality, provide customer satisfaction and considerably reduce overall cost, time and initiation of further work.
- (b) In order to minimize efforts in defect prediction with more accuracy and higher efficiency, it necessitates identifying newer methods and datasets by applying more sophisticated methodologies which will be appropriate and have adequate positive and effective impact on prediction of software defects.
- (c) Although considerable work has been made so far for prediction of software defects by applying various parameters, but it may be safely stated that sufficient work had not yet been done in defect prediction of the wave applications and open source software. As such, there is a need for further research work to find out more effective methodologies that may produce better result with higher accuracy in case of predicting software defects.

5 Challenging Issues

After critical analysis, various challenging issues have come to the light that requires immediate attention and timely solution. Owing to various reasons, application of methodologies is not totally problem or defect-free. In-fact, most of the studies implemented open source or public datasets and so, they may not work effectively for private and commercial datasets. Moreover, owing-to privacy issues, the proprietary datasets are not available in public. If availability of proprietary datasets is more, then it may help cross-project defect prediction with higher accuracy. Although various open or public datasets are available for defect prediction but each dataset is not having same number of metrics and similar type of metrics. These metrics are evaluated from different domain and the defect prediction model based object-oriented metrics is not applicable for different metrics or different feature-space. That-is-why, cross-project defect prediction is not very easy and feasibility of cross-project defect prediction model being wide acceptable is very less. It has however been accepted that this model is very useful for the industry. Various defect prediction models that have been proposed so far, could not at all give any guarantee for result of prediction. It is essential to undertake further studies on new metrics, new model or new development process that may be better performance, result-orientated and widely acceptable.

6 Conclusion and Future Work

Defect prediction in software system is truly crucial since, it is considered as an important step for enhancing software quality. Defect prediction in software system with application of proper methodologies is truly significant as it may immensely help in directing test efforts, reducing costs and improving quality and reliability of software. Research work in this field has emerged since 1992 and having huge volume of work done during last 25 years or so, but still it lacks in some areas and needs to solve those issues. However, unique and updated works (from the year 2005 to 2015) have been analyzed separately and the findings reveal that particularly Advance Machine Learning (AML), Neural Network (NN) and Support Vector Machine (SVM) methodologies are the most frequently used techniques as compared to all other techniques for predicting defects of software system. Moreover, it was also an important observation that public datasets used for this purpose comprise 64.79 % where as studies using private datasets cover only 35.21 %. We may conclude by stating that though different methodologies have been applied but no single methodology can be considered as a full proof for predicting software defects. It is highly essential to undertake further work applying newer methodologies in the initial stage for defect prediction with special emphasize on public datasets that are better result-orientated with higher level of accuracy. This work will facilitate further work and make endeavors in designing newer metrics of software that would pave the way and have all the potential to achieve higher prediction accuracy.

References

1. G. Abaei, A. Selamat, A survey on software fault detection based on different prediction approaches. *Vietnam J. Comput. Sci.* **1**(2), 79–95 (2014)
2. G. Abaeia, A. Selamata, H. Fujita, An empirical study based on semi-supervised hybrid self-organizing map for software fault prediction. *Knowl.-Based Syst.* **74**, 28–39 (2015)
3. A. Adline, M. Ramachandran, Predicting the software fault using the method of genetic algorithm. *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.* **3**(2), 390–398 (2014)
4. S. Agarwal, D. Tomar, A feature selection based model for software defect prediction. *Int. J. Adv. Sci. Technol.* **6**(5), 39–58 (2014)
5. E. Ardil, E. Ucar, P.S. Sandhu, Software maintenance severity prediction with soft computing approach. *Int. Sch. Sci. Res. Innov. Proc. World Acad. Sci. Eng. Technol.* **3**(2), 253–258 (2009)
6. M.M. Askari, V.K. Bardsiri, Software defect prediction using a high performance neural network. *Int. J. Softw. Eng. Appl.* **8**(12), 177–188 (2014)
7. M. Banga, Computational hybrids towards software defect predictions. *Int. J. Sci. Eng. Technol.* **2**(5), 311–316 (2013)
8. G.D. Boetticher, Nearest Neighbour Sampling for Better Defect Prediction. *ACM Journal* **30**(4), 1–6 (2005)
9. A. Campan, G. Serban, T.M. Truta, A. Marcus, An algorithm for the discovery of arbitrary length ordinal association rules. *DMIN* **6**, 107–113 (2006)
10. C. Catal, U. Sevim, B. Diri, Practical development of an eclipse-based software fault prediction tool using Naive Bayes algorithm. *Expert Syst. Appl.* **38**(3), 2347–2353 (2011)
11. C. Catal, Software fault prediction: a literature review and current trends. *Expert Syst. Appl.* **38**(4), 4626–4636 (2011)
12. C.P. Chang, C.P. Chua, Y.F. Yeh, Integrating in process of software defect prediction with association mining to discover defect pattern. *Inf. Softw. Technol.* **51**(2), 375–384 (2009)
13. Y. Chen, P. Du, X.H. Shen, P. Du, B. Ge, Research on software defect prediction based on data mining, computer and automation engineering ICCAE, in *The 2nd International Conference*, vol. 1, pp. 563–567 (2010)
14. A. Chug, S. Dhall, Software defect prediction using supervised learning algorithm and unsupervised learning algorithm, confluence 2013, in *The Next Generation Information Technology Summit (4th International Conference)*, pp. 173–179 (2013)
15. P.M. Dhiman, R. Chawla, A clustered approach to analyze the software quality using software defects, advanced computing & communication technologies ACCT, in *2nd International Conference*, pp. 36–40 (2012)
16. E. Erturk, E.A. Sezer, A comparison of some soft computing methods for software fault prediction. *Expert Syst. Appl.* **42**(4), 1872–1879 (2015)
17. K. Gao, T.M. Khoshgoftaar, Software defect prediction for high-dimensional and class-imbalanced data, in *Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering SEKE* (2011)
18. M. Gayathri, A. Sudha, Software defect prediction system using multilayer perceptron neural network with data mining. *Int. J. Recent Technol. Eng.* **3**(2), 54–59 (2014)
19. N. Gayatri, S. Nickolas, A.V. Reddy, Feature selection using decision tree induction in class level metrics dataset for software defect predictions. *Proc. World Congr. Eng. Comput. Sci.* **1**, 1–6 (2010)
20. D. Gray, D. Bowes, N. Davey, Y. Sun, B. Christianson, Software defect prediction using static code metrics underestimates defect-proneness, in *International Joint Conference on Neural Network IJCNN*, pp. 1–7 (2010)
21. D. Gray, D. Bowes, N. Davey, Y. Sun, B. Christianson, Using the support vector machine as a classification method for software defect prediction with static code metrics. *Eng. Appl. Neural Netw.* **43**, 223–234 (2009)

22. T. Hall, S. Beecham, D. Bowes, D. Gray, S. Counsell, A systematic literature review on fault prediction performance in software engineering. *IEEE Trans. Softw. Eng.* **38**(6), 1276–1304 (2012)
23. Z. Jianhong, P.S. Sandhu, S. Rani, A neural network based approach for modelling of severity of defects in function based software systems. *Int. Conf. Electron. Inf. Eng. ICEIE* **2**, 568–575 (2010)
24. X.Y. Jing, Z.W. Zhang, S. Ying, Y.P. Zhu, F. Wang, Software defect prediction based on collaborative representation classification, in *Proceedings in ICSE Companion, Proceedings of the 36th International Conference on Software Engineering*, pp. 632–633 (2014)
25. C. Jones, O. Bonsignour, *The Economics of Software Quality* (Pearson Education Inc., 2012)
26. Y. Kamei, A. Monden, S. Morisaki, K.I. Matsumoto, A hybrid faulty module prediction using association rule mining and logistic regression analysis, in *Proceedings of Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 279–281 (2008)
27. A. Kaur, I. Kaur, Empirical evaluation of machine learning algorithms for fault prediction. *Lect. Notes Softw. Eng.* **2**(2), 176–180 (2014)
28. J. Kaur, P.S. Sandhu, A K-Means based approach for prediction of level of severity of faults in software systems, in *Proceedings of International Conference on Intelligent Computational Systems* (2011)
29. K. Kaur, Analysis of resilient back-propagation for improving software process control. *Int. J. Inf. Technol. Knowl. Manage.* **5**(2), 377–379 (2012)
30. S. Kaur, D. Kumar, Software fault prediction in object oriented software systems using density based clustering approach. *Int. J. Res. Eng. Technol. IIRET* **1**(2), 111–116 (2012)
31. I.H. Laradji, M. Alshayeb, L. Ghouti, Software defect prediction using ensemble learning on selected features. *Inf. Softw. Technol.* **58**, 388–402 (2015)
32. M. Li, H. Zhang, R. Wu, Z.H. Zhou, *Sample-based Software Defect Prediction with Active and Semi-Supervised Learning, Automated Software Engineering*, vol. 9, no. 2 (Springer Publication, 2011), pp. 201–230
33. R. Li, S. Wang, An empirical study for software fault-proneness prediction with ensemble learning models on imbalanced data sets. *J. Softw.* **9**(3), 697–704 (2014)
34. Y. Liu, T.M. Khoshgoftaar, N. Seliya, Evolutionary optimization of software quality modelling with multiple repositories. *IEEE Trans. Softw. Eng.* **36**(6), 852–864 (2010)
35. R. Malhotra, Comparative analysis of statistical and machine learning methods for predicting faulty modules. *ELSEVIER J. Appl. Soft Comput.* **21**, 286–297 (2014)
36. T. Menzies, Z. Milton, B. Turhan, B. Cukic, Y. Jiang, A. Bener, *Defect Prediction from Static Code Features: Current Results, Limitations, New Approaches, Automated Software Engineering*, vol. 17, no. 4, pp. 375–407 (2010)
37. H. Najadat, I. Alsmadi, Enhance rule based detection for software fault prone modules. *Int. J. Softw. Eng. Appl.* **6**(1), 75–86 (2012)
38. A. Okutan, O.T. Yildiz, A novel regression method for software defect prediction with kernel methods, in *International Conference on Pattern Recognition Applications and Methods ICPRAM*, pp. 216–222 (2013)
39. J.C. Riquelme, R. Ruiz, D. Rodriguez, J.S. Anguilar-Ruiz, Finding defective software modules by means of data mining methodologies. *Latin Am. Trans. IEEE* **7**(3), 377–382 (2009)
40. K. Sankar, S. Kannan, P. Jennifer, Prediction of code fault using Naive Bayes and SVM classifiers. *Middle-East J. Sci. Res.* **20**(1), 108–113 (2014)
41. L. Sehgal, N. Mohan, P.S. Sandhu, Quality prediction of function based software using decision tree approach, in *International Conference on Computer Engineering and Multimedia Technologies*, pp. 43–47 (2012)
42. P.A. Selvaraj, P. Thangaraj, Support vector machine for software defect prediction. *Int. J. Eng. Technol. Res.* **1**(2), 68–76 (2013)
43. Q. Song, Z. Jia, M. Shepperd, S. Ying, J. Liu, A general software defect-proneness prediction framework. *IEEE Trans. Softw. Eng.* **37**(3), 356–370 (2011)

44. Q. Song, M. Shepperd, M. Cartwright, C. Mair, Software defect association mining and defect correction effort prediction. *IEEE Trans. Softw. Eng.* **32**(2), 69–82 (2006)
45. X. Tan, X. Peng, S. Pan, W. Zhao, Assessing software quality by program clustering and defect prediction, in *18th Working Conference on reverse Engineering* (2011)
46. W. Tao, L. Wei-Hua, Naive Bayes software defect prediction model, in *International Conference on Computational Intelligence and Software Engineering*, pp. 1–4 (2010)
47. J. Xu, D. Ho, L.F. Capretz, An empirical study on the procedure to derive software quality estimation models. *Int. J. Comput. Sci. Inf. Technol. IJCSIT* **2**(4), 1–16 (2010)
48. X. Yang, K. Tang, X. Yao, A learning-to-rank approach to software defect prediction. *IEEE Trans. Reliab.* **64**(1), 234–246 (2014)
49. W. Zhang, Y. Yang, Q. Wang, Using Bayesian Regression and EM Algorithm with Missing Handling for Software Effort Prediction. *Inf. Softw. Technol.* **58**, 58–70 (2015)