A Review of Software Defect Prediction Models

Harshita Tanwar and Misha Kakkar

Abstract This paper analyzes the performance of various software defects prediction techniques. Different datasets have been analyzed for finding defects in various researches. The main aim of this paper is to study many techniques used for predicting defects in software.

Keywords Defect prediction models · Redundant metrics · Attribute selection process · Software quality

1 Introduction

As use of software is increasing in various fields such as hospital, IT companies, banking, etc. So, having defects free software is very important. A high quality of software can be obtained by using SDP model. SDP models identify the bugs in the particular software at the early stage that is at the stage of software development. This SDP model is trained with the help of software metrics or attributes. Effectiveness of SDP is based on the characteristics of various metrics of a particular software. These metrics are used to find whether a software contains the defective modules or not. Researches are done regarding selection of attributes in order to develop as much as effective SDP model.

To construct effective software defect prediction model first data is collected and then, analyzed. Many techniques can be used for preprocessing of data which includes data cleaning, feature selection, variable clustering, VIF, Spearman, redundant analyses etc. Datasets from these preprocessing techniques are then used for training SDP models. For constructing SDP models, many algorithms such as

H. Tanwar (✉) · M. Kakkar
Department of CSE, Amity University, Sec-125, Noida, Uttar Pradesh, India
e-mail: tanwar.harshita92@gmail.com

M. Kakkar
e-mail: mkakkar@amity.edu

V. E. Balas et al. (eds.), *Data Management, Analytics and Innovation*,
Advances in Intelligent Systems and Computing 808,
https://doi.org/10.1007/978-981-13-1402-5_7

KNN, NN, SVM, Naïve Bayes and random forest can be used. Prediction output then determines whether the dataset contains defect metrics or not.

The performances of these SDP models can be evaluated using performance indicator that is CA (Classifier Accuracy), AUC (area under curve), Precision and Recall etc. Also many SDP models such as random forest, fuzzy logic system, SAL, regression analyses etc. are introduced by researchers.

This review paper is organized as follows: Sect. 2 consists of review procedure part, Sect. 3 contains literature review part, Sect. 4 contains the conclusion part and last section contains the references.

2 Review Procedure

In order to analyze the performance of various SDP models, we have reviewed 20 relevant research papers out of 100 research paper. We find the relevant paper for review based on the following steps:

(i) Downloaded the research paper using the search keywords: Software Defect Prediction.
(ii) Read the title, Abstract and conclusion of research papers.
(iii) Selected the 20 relevant paper after reading the content of 100 research paper.
(iv) Results and conclusion of 20 paper is then analyzed thoroughly.

Figure 1 describes the flowchart used for defect prediction.

To analyze SDP, we formulate the following research questions to keep review focused

RQ1: what are the different techniques of software defects prediction?
RQ2: what are the measures that effect the performance of SDP models?
RQ3: How irrelevant data can introduce defects in software?
RQ4: what methods can be used for improving software defects prediction models?

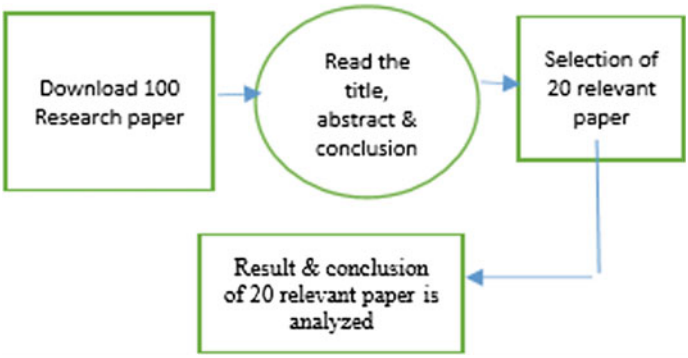

Fig. 1 Flowchart for review process

3 Literature Review

It has been analyzed from review of 20 research papers that mainly three techniques are used for implementing the SDP models that is classification, regression and clustering. Many researches on SDP model done by researcher are discussed below:

In [1], Ai-jamimi and Hamid proposed a fuzzy logic-based SDP model. The performance of this logic-based prediction model has been checked by real software projects data. They find this model as the most effective way to obtain dominant set of metrics. This in turn make fuzzy logic-based model more valid and satisfactory as compared to other models. Result showed that using all software metrics gives the lowest accuracy and less satisfaction as compared with the other set of metrics. The relevant set of metrics gives better result that is metrics obtained after removal of redundant metrics.

In [2], Koroglu et al. used seven old versions of software and their additional feature to find the defects of current versions. They compared several SDP process that is Naïve Bayes, decision tree, and random forest and finds the random forest has the highest predictive power as compared to other models. All these models are compared with the AUC value that is area under curve. They find that random forest has the highest AUC value.

In [3], Sharmin proposed a novel technique of attribute selection that is selection of attribute with log filtering (SAL). They used the log filtering to preprocess the data. Finally, comes to the conclusion that this method gives the more accuracy of SDP as compared to other techniques. This method is applied on several widely used publicly available datasets

In [4], Sethi and Gagandeep find that the artificial neural network (ANN) gives the better result as compared to fuzzy based logic model. ANN gives the more accurate value. It can be used in hybrid approach to a large dataset. These model is analyzed with the mean magnitude of relative error (MMRE) and balanced mean magnitude of relative error (BMMRE).

In [5], Suffian used the metrics in order to find the performance of different models that is regression model with other models. They find that regression analysis is most accurate as compared to other models. They used the p-value of 0.05 as the threshold for the selection of attributes of software.

In [6], Ami et al. proposed a novel approach of attribute selection method for construction of effective defect prediction model. This approach finds the attributes with high accuracy by calculating the total weight of each attribute and sorting each attribute based on total weight. They used the one classifier that is Naïve Bayes in their study in order to construct the SDP model.

In [7], Can et al. introduced a novel approach for software defect prediction PSO and SVM called as P-SVM model and observed that P-SVM has more accuracy than BP neural network, SVM Model and GA-SVM model. They found this model as most robust. The dataset used is only JM1 for proposing the novel approach of P-SVM.

In [8], Jiarpakdee finds after studying 101 available datasets that 10–67% of metrics of these datasets are redundant. Also, it has been observed that elimination of redundant metrics before constructing the SDP model is very important. It improves the performance of SDP model.

In [9], Wang et al. observed that multivariant Gauss Naïve Bayes has best performance as compared to all kind of classifiers. It is most effective defect prediction model. They also experiment with J48 in order to find the performance of multivariant Gauss Naïve Bayes. They found that MVGNB is most effective in predicting the defects at an early stage of software development.

In [10], Liu et al. proposed a SDP model for that service oriented software. They find the SDP model based on the present model, QDPSOMO. It provides better management of quality for software that depends on EXPERT COCOMO. It is formed by the combination of defect prediction, measurement and management.

In [11], Kakkar and Sarika Jain concluded from their research work that hybrid model of classifier or the combination of one or more classifier always gives the better result than any single classifier. The hybrid approach of selection of attribute gives more accuracy. It also helps us to analyze the impact of attribute selection and preprocessing of data on different SDP models. Performance of five classifiers has been compared, i.e., IBk, KStar, LWL, Random forest, and Random tree. It has been observed that LWL gave the accuracy of 92.23% and has best performance.

In [12], Verma and Kumar analyzed the multiple regression in their research work. They find the impact of clustering on defect prediction. Three clusters are formed. Result has shown that prediction model formed after clustering showed better result rather than applying prediction model on whole software project.

In [13], Yang et al. proposed a novel approach that is learning-to-rank (LTR) approach for the construction of SDP model. This approach helps to find the test resources more effectively by finding which module of software have more defects. They found that learning to rank approach gives better prediction accuracy as compared to linear model using LS. However, LTR in some cases is not giving as better result as given by Random Forest. LTR is not performing better in all cases.

In [14], Sawadpong and Allen use a exceptional handling for implementation of SDP model. They proposed exception-based software metrics. It is based on the structural attributes of exception handling call graphs. They came to the conclusion that if SDP model that is depends on exceptional based metrics gives more result as compared to conventional prediction model. They used the software repositories that have mined data and defect reports for their research.

In [15], Shuai et al. implemented Genetic algorithm with SVM (GA-CSSVM) on NASA datasets. They concluded that GA-CSSVM performed better as compared to increases normal SVM.

In [16], Gabriel Kofi Armah et al. performed Multilevel preprocessing by selecting the attributes twice and filtering instance thrice. Four K-NN classifier's preprocessing that is KNN-LWL, KStar, IBK, and IB1 results were analyzed and compared with random tree, random forest, and non-nested generalized classifier. Four performance parameter that is accuracy, recall, Area under curve (AUC) and

Table 1 Summary of studied research papers

S. No.	Title	Authors	Year and publication	Basic techniques	Dataset used	Methods for improving
1	Toward comprehensible SDP models using fuzzy logic	Ai-jamimi and Hamid	2016, IEEE	Fuzzy logic-based software prediction model	PROMISE data repository	Attribute selection using trapezium and triangular membership's functions of metrics
2	Defect prediction on a legacy industrial software : a case study on software with few defects	Y. Koroglu et al.	2016, ACM	Random forest	Data collected JIRA entries of previous seven older version of software	Ranking metrics using Information gain
3	Improved approach for SDP using artificial neural networks	T. Sethi et al.	2016, IEEE	ANN based techniques	20 genuine software venture datasets	Fuzzy logic-based approach for metrics evaluation
4	A study of redundant metrics in defect prediction datasets,	Jiarpakdee et al.	2016, IEEE	Analyze how redundant metrics effects the performance of SDP model	NASA defect datasets	1. Redundancy analysis 2. Spearman method 3. Variable clustering
5	Feature selection in SDP: a comparative study	Kakkar et al.	2016, IEEE	IBk, LWL, k-star, Random forest and random tree	NASA MDP datasets named CM1, JM1, KC1, KC3 and PC1	Hybrid attribute selection approach
6	SDP using exception handling call graphs : a case study	P. Sawadpong et al.	2016, IEEE	Exception-based software metrics SDP model	Defect data from Hadoop0.19.0 and hadoop0.20.20	Exceptional handling call graphs
7	Emperical study of defects dependency on software metrics using clustering approach	D. K. Verma et al.	2015, IEEE	Regression technique	NASA PC1 DATASETS	Clustering based metrics selection method
8	A learning-to-rank approach to software defect prediction	X. Yang et al.	2015, IEEE	Proposed a novel approach that is learning to rank (LTR) approach for the construction of SDP model	Eclipse datasets	Info gain metrics selection method

(continued)

Table 1 (continued)

S. No.	Title	Authors	Year and publication	Basic techniques	Dataset used	Methods for improving
9	An effective method for software defect prediction	S. Sharmin et al.	2015, IEEE	Proposed a novel technique of attribute selection	NASA datasets	SAL (Selection of attribute with log filtering)
10	Selecting best attributes for software defect prediction	Ami et al.	2015, IEEE	Finds the way of attribute selection such that performance of SDP increases	NASA datasets	Naïve Bayes classifier
11	A new model for software defect prediction using PSO and SVM	He Can et al.	2013, IEEE	Introduced a novel approach for SDP model using PSO and SVM called as P-SVM model	NASA dataset named JM1	Optimization theory
12	SDP using dynamic support vector machine	B. Shuai	2013, IEEE	Naïve Bayes, MLP, J48, Random Forest	NASA datasets CM1, KC1, PC1, JM1	Accuracy increases with feature selection but decreases at further stage as important features are lost
13	Multilevel data preprocessing for software defect prediction	Gabriel Kofi Armah et al.	2013, IEEE	KNN (LWL, KStar, IBk, IB1), non-nested generalized exemplars (NNGE), random tree and random forest	NASA Datasets CM1, PC1 JM1, KC2 KC1	Double selection of attributes and triple filtering of instances gives better accuracy than classifying training set directly
14	Assuring software quality using data mining methodology: a literature study	Arun Singh et al.	2013, IEEE	LR, Random forest, SVM, fuzzy programming association rule mining, NB, ANN and genetic algorithm	NA	Mining techniques help eliminate vestigial defects
15	A prediction model for system testing defects using regression analysis	M. Suffian	2012, International Journal of Soft Computing and Software Engineering	Analyze the performance of different model using metrics	Data is based on the software development using V-shape development model	Statistical approach

(continued)

Table 1 (continued)

S. No.	Title	Authors	Year and publication	Basic techniques	Dataset used	Methods for improving
16	A data-driven model for software reliability prediction	Jung-Hua Lo et al.	2012, IEEE	ARIMA and SVM	Dataset1: project from Rome air development Centre Dataset2: Project given by Hu et al.	Hybrid model performs better in SDP and decreases error rate
17	Naïve Bayes software defect prediction model	T. Wang and W. Li	2010, IEEE	Introduced machine learning algorithm for implementing SDP	NASA datasets named CM1, JM1, KC1, KC2 and PC1	Multivariants Gauss Naïve Bayes
18	A defect prediction model for software based on service oriented architecture using EXPERT COCOMO	Jun Liu et al.	2009, IEEE	Gives a SDP model for service oriented software	Genuine software	Used software that depends on Expert COCOMO
19	Defect prediction for embedded software	Ataç Deniz Oral et al.	2007, IEEE	MLP, Naïve Bayes, classification by voting feature intervals (VFI)	NASA datasets named CM1, PC1, PC3, PC4	Ensemble proves to be the best performer with 73% balance
20	Empirical assessment of machine learning based SDP techniques	Venkata et al.	2005, IEEE	Linear Regression, support vector logistic regression, support vector regression, pace regression, 1R, neural network for continuous and discrete gold field, NB, instance based learning, J48	NASA datasets named CM1, PC1, JM1, KC1	NB, IBL and NN perform better than other prediction models, also NB alone was best performer for 3 datasets

precision are used to compare them. Results showed that performance of Random Forest increased by performing double preprocessing.

In [17], Lo et al. combined SVM and Auto Regression Integrated Moving Average (ARIMA) for SDP. They analyzed that performance of hybrid model is better as compared to conventional prediction model and decreases error rate.

In [18], Oral et al. performed SDP by combining three classification techniques that is NB, voting feature interval and MLP using five datasets. He concluded that combination of these classifiers gives better performance to SDP models especially for embedded system.

In [19], Singh et al. analyzed the performance of different mining techniques that is Logistic Regression, random forest, C4.5, Association Rule Mining, Naïve Bayes, ANN, SVM, genetic algorithm and Fuzzy Programming. They concluded that Data Mining techniques are very helpful for removing minor defects.

In [20], Challagulla et al. compared 13 machine learning methods. They find that NB, neural network, and Instance-based learning performed better than other as compared to all other methods.

As seen from Table 1, there are many techniques use for the implementation of SDP models. Some of these techniques are fuzzy logics based, ANN based model, P-SVM model, Multivariant Gauss Naïve Bayes model, random forest method, regression analysis and many more.

NASA datasets are the most commonly used dataset for analyses of defects in software.

4 Conclusion

There are many techniques for constructing SDP models such as fuzzy logic-based software prediction, Naïve Bayes, neural network, random forest, SVM, P-SVM, etc. Different researcher performs preprocessing with different techniques and comes out with different conclusions. It has been observed that selection of attributes effects the performance of SDP model. There are many measures that effect the performance of SDP models that is AUC (area under curve), precision, recall, classifier accuracy, etc. However, introduction of irrelevant data decreases the performance of SDP model. Many methods are there for improving the performance of SDP that is multiple regression, multivariant Naïve Gauss Bayes, Info gain metrics selection method, SAL (Selection of attribute using log filtering), statistical approach, optimization theory, Exceptional handling call graphs etc. Based on the analysis, further new techniques can be introduced for constructing the better SDP models.

References

1. Ai-jamimi, H. A. (2016). Toward comprehensible software defect prediction models using fuzzy logic (pp. 127–130).
2. Koroglu, Y., Sen, A., Kutluay, D., Bayraktar, A., Tosun, Y., Cinar, M., & et al. (2016). Defect prediction on a legacy industrial software : A case study on software with few defects. In *2016 IEEE/ACM 4th International Workshop on Conducting Empirical Studies in Industry (CESI)* (pp. 14–20).
3. Sharmin, S. (2015). SAL: An effective method for software defect prediction (pp. 184–189).
4. Sethi, T., & Gagandeep. (2016). Improved approach for software defect prediction using artificial neural networks. In *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)* (pp. 480–485).
5. Suffian, M. D. M., Ibrahim, S., Dhiauddin, M., Suffian, M. D. M., & Ibrahim, S. (2012). A prediction model for system testing defects using regression analysis. *International Journal of Soft Computing and Software Engineering, 2*(7), 69–78.
6. Mandal, P., & Ami, A. S. (2015). Selecting best attributes for software defect prediction. In *2015 IEEE International WIE Conference on Electrical and Computer Engineering* (pp. 110–113).
7. Can, H., Jianchun, X., Ruide, Z., Juelong, L., Qiliang, Y., & Liqiang, X. (2013). A new model for software defect prediction using Particle Swarm Optimization and support vector machine. In *2013 25th Chinese Control and Decision Conference* (pp. 4106–4110).
8. Jiarpakdee, J., Tantithamthavorn, C., Ihara, A., & Matsumoto, K. (2011). A study of redundant metrics in defect prediction datasets (pp. 37–38).
9. Wang, T., & Li, W. (2010). Naïve Bayes software defect prediction model. *IEEE*, no. 2006 (pp. 0–3).
10. Liu, J., Xu, Z., Qiao, J., & Lin, S. (2009). A defect prediction model for software based on service oriented architecture using EXPERT COCOMO. In *2009 Chinese Control and Decision Conference* (pp. 2591–2594).
11. Kakkar, M., & Jain, S. (2016, January). Feature selection in software defect prediction: A comparative study. In *2016 6th International Conference on Cloud System and Big Data Engineering (Confluence),* (pp. 658–663).
12. Verma, D. K., & Kumar, S. (2015). Emperical study of defects dependency on software metrics using clustering approach (pp. 0–4).
13. Yang, X., Tang, K., & Yao, X. (2015). A learning-to-rank approach to software defect prediction. *IEEE Transactions on Reliability, 64*(1), 234–246.
14. Sawadpong, P., & Allen, E. B. (2016). Software defect prediction using exception handling call graphs : A case study.
15. Shuai, B., Li, H., Li, M., Zhang, Q., & Tang, C. (2013). Software defect prediction using dynamic support vector machine. In *2013 9th International Conference on Computational Intelligence and Security (CIS)* (pp. 260–263).
16. Armah, G. K., Luo, G., & Qin, K. (2013). Multi_level data pre_processing for software defect prediction. In *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII) (*pp. 170–174).
17. Lo, J.-H. (2012). A data-driven model for software reliability prediction. In *IEEE International Conference on Granular Computing*.
18. Oral, A. D., & Bener, A. B. (2007, November). Defect prediction for embedded software. In *22nd International Symposium on Computer and Information Sciences, 2007. ISCIS 2007* (pp. 1–6). New York: IEEE.
19. Singh, A., & Singh, R. (2013, March). Assuring Software Quality using data mining methodology: A literature study. In *2013 International Conference on Information Systems and Computer Networks (ISCON)* (pp. 108–113). New York: IEEE.
20. Challagulla, V. U. B., Bastani, F. B., Yen, I. L., & Paul, R. A. (2008). Empirical assessment of machine learning based software defect prediction techniques. *International Journal on Artificial Intelligence Tools*, *17*(02), 389–400.