World Journal of Engineering Research and Technology

WJERT

www.wjert.org SJIF Impact Factor: 5.924

# A REVIEW ON SOFTWARE DEFECT PREDICTION USING DATA MINING TECHNIQUES

***[1]Rakesh Kumar and [2]Dr. Dharmendra Chourishi**

M. Tech. Scholar, CSE, NRI College Bhopal A.P., NRI College Bhopal.

**Article Received on 01/11/2019** **Article Revised on 21/11/2019** **Article Accepted on 22/12/2019**

***Corresponding Author**
**Rakesh Kumar**
M. Tech. Scholar, CSE,
NRI College Bhopal A.P.,
NRI College Bhopal.

## ABSTRACT

The common software problems appear in a wide variety of applications and environments. Some software related problems arises in software project development i.e. software related problems are known as software defect in which Software bug is a major problem arises in the coding implementation. Software defect prediction has been one of the key areas of exploration in the domain of software quality. The ability of a model to learn from data that does not come from the same project or organization will help organizations that do not have sufficient training data or are going to start work on new projects. The findings of this research are useful not only to the software engineering domain, but also to the empirical studies, which mainly focus on symmetry as they provide steps-by-steps solutions for questions raised in the article. A typical software development process has several stages; each with its own significance and dependency on the other. Each stage is often complex and generates a wide variety of data. Using data mining techniques, Hidden patterns can be uncovered from this data, which measure the impact of each stage on the other and gather useful information to improve the software development process. The insights gained from the extracted knowledge patterns can help software engineers to predict, plan and comprehend the various intricacies of the project, allowing them to optimize future software development activities. As every stage in the development process entails a certain outcome or goal, it becomes crucial to select the best data mining techniques to achieve these goals efficiently.

**KEYWORDS:** Classification, Software BUG, defect, machine learning, verification; prediction, software metrics, security, Data Mining, Software Engineering, Software Development.

## 1. INTRODUCTION

Defect Prediction in Software (DeP) is the process of determining parts of a software system that may contain defects.[1] Application of DeP models early in the software lifecycle allows practitioners to focus their testing manpower in a manner that the parts identified as "prone to defects" are tested with more rigor in comparison to other parts of the software system.[2] This leads to the reduction of manpower costs during development and also relaxes the maintenance effort.[3,4] DeP models are built using two approaches: First, by using measurable properties of the software system called Software Metrics and second, by using fault data from a similar software project. Once built, the DeP model can be applied to future software projects and hence practitioners can identify defects prone parts of a software system. Initially, the models used for DeP were built using statistical techniques, but to make the model intelligent, i.e., capable of adapting to changing data in such a manner that as the development process matures the DeP model also matures; it is important that learning techniques are used while building DeP models. In the past, a large number of DeP studies have made use of machine learning techniques. some authors used Decision Tree (DT),[5] and Artificial Neural Network (ANN),[6] to build DeP models using Object-Oriented (OO) metrics.[7]

### 1.1 Data Mining

It is processes in computer science by which relationship and pattern from data can be easily extracted and information is collected that helps in decision making as required in software development field. It easily extracts information from problem reports and take decision by the help of information software defect can detect and software quality is improved.

### 1.2 Classification

Classification have a training set which provide a facility to have a common level of same classes of data. Some different type bugs in software project development: SW-bug, document bug, duplicate bug and mistaken bug. These bugs have common level bug classes of data object known as software defect in training set.

### 1.3 Decision Tree

Decision tree is a classifier of root node which generates other branches as a node. The common attributes of data at class level each node have own information.

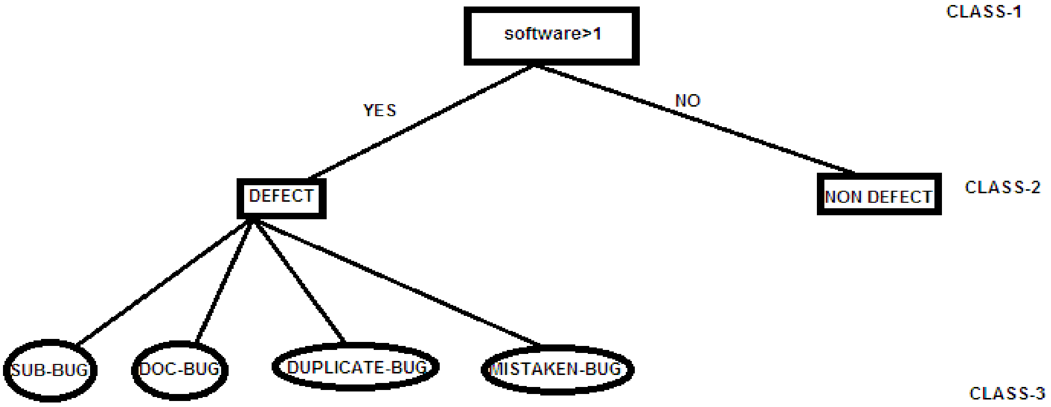

**Fig 1: Represents to check level defect of software.**

**For example**

1. **If software > 1** then root node extract another branches or internal node (not leaf node) show class.[2]
2. **If software < 1** then shows root node on class.[1]
3. **If software defect > 1** then found defect classification categories bug at class.[3] not extract another node otherwise on the class.[2]

### 1.4 Bayes Rule

Bayes rule have event and supporting evidence and there are two cases arise:

1. If event occurs means between evidence **P (H**) probability conform.
2. Event occurs means with supporting evidence **P (H/E).**

Let **H** be the event of SW-bug and **E** be the evidence of software defect then we have P (SW-bug/software defect) =P (software defect/SW-bug)*P (SW-bug)/P (software defect) Naïve Bayes classification algorithm basically used for high dimension input from the above example. We can predict and output of some event and observing some evidence. Generally it is better to have more than one evidence to support the prediction of an event.

## 2. LITERATURE REVIEW

Shepperd, Schofield and Kitchenham,[8] discussed that need of cost estimation for management and software development organizations and give the idea of prediction and discuss the methods for estimation.

Alsmadi and Magel,[9] discussed that how data mining provide facility in new software project its quality, cost and complexity also build a channel between data mining and software engineering.

Boehm, Clark, Horowitz, Madachy, Shelby and Westland,[10] discussed that some software companies suffer from some accuracy problems depend on his data set after prediction software company provide new idea to specify project cost schedule and determine staff time table.

Pal and Pal,[11] conducted study on the student performance based by selecting 200 students from BCA course. By means of ID3, c4.5 and Bagging they find that SSG, HSG, Focc, Fqual and FAIn were highly correlated with the student academic performance.

K.Ribu,[12] discussed that the need of open source code projects analyzed by prediction and get estimating object oriented software project by case model.

Nagwani and Verma,[13] discussed that the prediction of software defect (bug) and duration similar bug and bug average in all software summery, by data mining also discuss about software bug.

Hassan,[14] discussed that the complex data source (audio, video, text etc.) need more of buffer for processing it does not support general size and length of buffer.

Chaurasia and Pal,[15,16] conducted study on the prediction of heart attack risk levels from the heart disease database with data mining technique like Naïve Bayes, J48 decision tree and Bagging approaches and CART, ID3 and Decision Table. The outcome shows that bagging techniques performance is more accurate than Bayesian classification and J48.

Li and Reformate,[17] discussed that the software configuration management is a system includes documents, software code, status accounting, design model defect tracking and also include revision data.

Elcan,[18] discussed that COCOMO model pruned accurate cost estimation and there are many thing about cost estimation because in project development involve more variable so COCOMO measure in term effort and metrics.

Chang and Chu,[19] discussed that for discovering pattern of large database and its variables also relation between them by association rule of data mining.

Kotsiantis and Kanellopoulos.[20] discussed that high severity defect in software project development and also discussed the pattern provide facility in prediction and associative rule reducing number of pass in database.

Pannurat, N. Kerdprasop and K. Kerdprasop,[21] discussed that association rule provide facility the relationship among large dataset as like software project term hug amount, cost record and helpful in process of project development.

Fayyad, Piatesky Shapiro, Smuth and Uthurusamy,[22] discussed that classification creates a relationship or map between data item and predefined classes.

Pal,[23] conducted study on the student dropout rate by selecting 1650 students from different branches of engineering college. In their study, it was found that student's dropout rate in engineering exam, high school grade; senior secondary exam grade, family annual income and mother's occupation were highly correlated with the student academic performance.

Shtern and Vassillios,[24] discussed that in clustering analysis the similar object placed in the same cluster also sorting attribute into group so that the variation between clusters is maximized relative to variation within clusters.

Runeson and Nyholm,[25] discussed that code duplication is a problem which is language independent. It is appear again and again another problem report in software development and duplication arises using neural language with data mining.

Primary studies by definition correspond to the literature being mappings. To provide a strong mapping it is essential that selection of primary studies for mapping must be done carefully. While it is good that an exhaustive search is conducted for the selection of primary studies, in some cases it is not possible because of the number of primary studies available. In such cases the search criteria become important. For DeP studies, we can conduct an exhaustive search because the number of primary studies is not very large and since we are only concerned with those studies which are empirical in nature, the number shrinks further. We have selected the list of following digital libraries to perform our search:

1. IEEE Xplore

2. Springer Link
3. Science Direct
4. Wiley Online Library
5. ACM Digital Library
6. Google Scholar

The Search String: A search string is the combination of characters and words entered by a user into a search engine to find desired results. The information provided to the search engine of the digital library directly impacts the results provided by it. To ensure that all the primary studies that our mapping plans to address are covered we need to be careful in the selection and placement of keywords used in the search string.

Search string. software.(defect + fault). (software metrics + object oriented metrics + design metrics) Here, '.' corresponds to the Boolean AND operation, and '+' Corresponds to the Boolean OR operation. The search string was executed on all six electronic databases mentioned above and the publication year was restricted to the range 1995–2018. The literature hence obtained was processed further using a carefully designed inclusion-exclusion criteria and quality analysis criteria as described in the following sections.

**The Inclusion-Exclusion Criteria:** The search results obtained by execution of the search string may still fetch some primary studies that either do not add value to the mapping or do not fall within the purview of what the mapping aims to accomplish.

Once all the primary studies have been obtained, a carefully designed inclusion-exclusion criteria are applied to the resultant set in order to eliminate entities that do not match the objectives of the mapping.

### Inclusion Criteria

1. Empirical studies for DeP using software metrics.
2. Studies that provide empirical analysis using statistical, search-based and machine learning techniques.

### Exclusion Criteria

- Literature Reviews and Systematic Reviews.
- Studies that do not use DeP as the dependent variable.

- Studies of non-empirical nature.
- If two studies by the same author(s) exist, where one is an extension of the previous work the former is discarded. But if the results differ, both are retained.

**Review Committee:** A review committee has been formed that comprises of two Assistant Professors and two senior researchers to mapping in order to rate all primary studies captured from the search. All studies were examined independently on the basis of the criteria defined in The Inclusion-Exclusion Criteria. Application of the inclusion-exclusion criteria resulted in 98 studies out of the total 156 studies being selected for quality analysis.

**Quality Analysis:** Assessing the quality of a set of primary studies is a challenging task. A quality analysis questionnaire is prepared as part of this systematic mapping to assess the relevance of studies taking part in this mapping. The questionnaire takes into consideration suggestions given in Reference.[26] A total of 18 questions, given in Table 1, together form the questionnaire and each question can be answered as "Agree" (1 point), "Neutral" (0.5 points) and "Disagree" (0 points). Hence, a study can have a maximum 18 points and minimum 0 points. The same review committee enforces the quality analysis questionnaire.

**Table 1: Quality analysis questions.**

| No. | Description | Agree | Neutral | Disagree |
|---|---|---|---|---|
| QQ1 | Is the objective of the study clear? | | | |
| QQ2 | Does the study add value to the existing literature? | | | |
| QQ3 | Is the dataset size sufficient for this type of studies? | | | |
| QQ4 | Does the study perform multi-co linearity analysis? | | | |
| QQ5 | Does the study perform feature sub-selection? | | | |
| QQ6 | Are the independent variables clearly defined? | | | |
| QQ7 | Is the data collection procedure clearly defined? | | | |
| QQ8 | Does the study use statistical tests while evaluating results? | | | |
| QQ9 | Does the author provide sufficient detail about the experiment? | | | |
| QQ10 | Are the threats to validity given? | | | |
| QQ11 | Does the study provide parameters of test? | | | |
| QQ12 | Are the limitations of the study given? | | | |
| QQ13 | Does the study clearly define the performance parameters used? | | | |
| QQ14 | Are the learning techniques clearly defined? | | | |
| QQ15 | Are the results clearly stated? | | | |
| QQ16 | Does the abstract provide sufficient information about the content of the study? | | | |
| QQ17 | Is there a comparison among techniques? ML (Machine Learning) vs. ML? | | | |
| QQ18 | Is there a comparison among techniques? ML vs. Statistical? | | | |

**Data Extraction:** To extract meaningful information for each study such that the research questions can be answered, the data extraction method should be objectively defined. We created the data form shown in Figure 3 and filled it for each of the primary study (PE) that passes the quality analysis criteria. The review committee filled the data extraction card and any conflicts raised during the data extraction process were resolved by taking suggestions from other researchers. The resultant data was converted to an Excel (.xlsx) Workbook for usage during the data synthesis process.

Author(s)
Title of Publication
Journal/Conference name
Year of Publication
Dataset(s) used
Independent Variables
Model Building Techniques Used
Feature Selection Techniques Used
Cross Project/Severity/Security Defects/Threshold Determining Studies
Research Questions Addressed (List)

**Figure 3: Extraction Card.**

## 3. Problems Identification And Objectives

### Problems

(i) Difficulty in separating correct theories from the incorrect ones when the purpose of evaluation is practiced.

(ii) Difficulty in the identification of quality literature from quality lacking literature.

### Objectives

1. The objective of this mapping is to establish a starting point for future research in DeP and simultaneously provide practitioners with a summary of all the work done in the area of DeP allowing them to pick a prediction policy that suits them.
2. To make extensive comparisons between search-based techniques, machine learning techniques and statistical techniques.

## 4. METHODOLOGY

The mapping method in this study is taken from Reference.[1] Figure 1 outlines the process diagram.

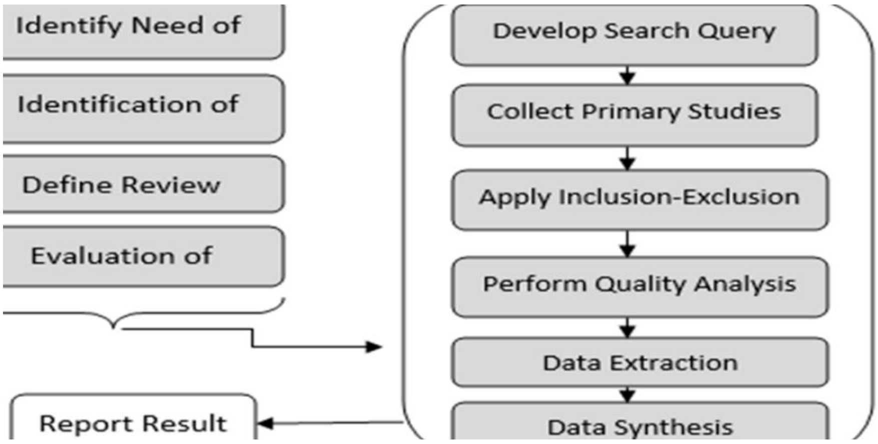

**Fig. 2: The mapping process.**

The first step is to identify the need for conducting the systematic mapping and establishing a protocol for the mapping. After this, research questions that the systematic mapping attempts to address are formulated. Once the questions have been identified and evaluated, a search query is designed which is used to extract studies from digital libraries. The studies collected are then passed through a four-stage process:

1. Application of Inclusion-Exclusion Criteria.
2. Quality Analysis of Included Studies.
3. Extraction of Data from selected Studies.
4. Identification of means to report results.

## CONCLUSIONS

In this survey, the need and importance of using data mining techniques has been established to aid software engineering, especially to tackle problems such as the occurrence of bugs, rise in the cost of software maintenance; unclear requirements, etc. that can affect software productivity and quality. This study has outlined the major research works that have taken place in this field. It have been also listed the sources of software engineering data that can be mined, most common stages in the development process as well as the data mining techniques that can be applied in these stages. However, the major contribution of this research work lies in the specification of the data mining technique most suited for a particular stage in the development process.

## REFERENCES

1. Alsmadi and Magel, “Open source evolution Analysis,” in proceeding of the 22nd IEEE International Conference on Software Maintenance (ICMS’06), phladelphia, pa. USA, 2006.
2. Approaches used in Software Engineering; The Irish Software Engineering Research Centre: Limerick.
3. Beecham, S.; Hall, T.; Bowes, D.; Gray, D.; Counsell, S.; Black, S. A Systematic Review of Fault Prediction.
4. Boehm, Clark, Horowitz, Madachy, Shelby and Westland, ”Cost models for future software life cycle Process: COCOMO2.0.” in Annals of software Engineering special volume on software process and prodocuct measurement, J.D. Arther and S.M. Henry, Eds, vol.1, pp.45-60, j.c. Baltzer AG, science publishers, Amsterdam, The Netherlands, 1995.
5. Catal, C.; Diri, B. A Systematic Review of Software Fault Prediction studies. Expert Syst. Appl, 2009: 36.
6. Chang and Chu, “software defect prediction Using international association rule mining”, 2009.
7. Chauraisa V. and Pal S., “Data Mining Approach to Detect Heart Diseases”, International Journal of Advanced Computer Science and Information Technology (IJACSIT), 2013; 2(4): 56-66.
8. Chauraisa V. and Pal S., “Early Prediction of Heart Diseases Using Data Mining Techniques”, Carib. j. Sci Tech, 2013; 1: 208-217.
9. Development effort estimation models. Inf. Softw. Technol, 2012; 54: 41–59. [CrossRef]
10. Elcan C., “The foundations of cost sensitive learning”, In processing of the 17 International conference on Machine learning, 2001.
11. Fawcett, T. An Introduction to ROC Analysis. Pattern Recognit. Lett., 2006; 27: 861–874. [CrossRef]
12. Fayyad, Piatesky Shapiro, Smuth and Uthurusamy, “Advances in knowledge discovery And data mining”, AAAI Press, 1996.
13. Hall, T.; Beecham, S.; Bowes, D.; Gray, D.; Counsell, S. Asystematic Literature Review on Fault Prediction.
14. Hampherey Watts S., “A discipline for software Engineering reading”, Ma, Addison Wesley, 1995.

15. Hassan, “The road ahead for mining software Repositories”, in processing of the future of software Hassan, A.E.; Holt, R.C. The top ten List: Dynamic Fault Prediction. In Proceedings of the 21st IEEE.
16. He, H.; Garcia, E.A. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering. IEEE Trans. Knowl. Data Eng, 2009; 21. [CrossRef]
17. In Proceedings of the IEEE 12th International Symposium on High Assurance Systems Engineering, 2010.
18. International Conference on Software Maintenance, Budapest, Hungary, September Ireland, 2010; 26–29.
19. J.R. Quinlan, “C4.5: programs for machine learning”, Morgan Kaufmann, San Francisco, 1993.
20. Kitchenham, B.A. Guidelines for Performing Systematic Literature Review in Software Engineering; Technical.
21. Kotsiantis and Kanellopoulos, “Associationn rule mining: A recent overview”, GESTS international transaction on computer science and Engineering, 2006.
22. Li and Reformat, “A practical method for the Software fault prediction”, in proceeding of IEEE Nation conference information reuse and Integration (IRI), 2007.
23. M. Shepperd, C. Schofield, and B. Kitchenham, ”Effort estimation using analogy,” in of the 18th International Conference On Software Engineering, pp.170- 178. Berlin, Germany, 1996.
24. Maintenance at the 24th IEEE international Conference on software maintenance, 2008.
25. Naeem Seliya, T.M.; Khoshgoftaar, J.; VanHulse, J. Predicting Faults in High Assurance Software.
26. Nagwani N. and Verma S., “Prediction data mining Model for software bug estimation using average Weighted similiarity,” In proceeding of advance Computing conference (IACC), 2010.
27. Pal A. K., and Pal S., “Analysis and Mining of Educational Data for Predicting the Performance of Students”, (IJECCE) International Journal of Electronics Communication and Computer Engineering, 2013; 4(5): 1560-1565, ISSN: 2278-4209.
28. Pal S., “Mining Educational Data to Reduce Dropout Rates of Engineering Students”, I.J. Information Engineering and Electronic Business (IJIEEB), 2012; 4(2): 1-7.
29. Pannurat, Kerdprasop and Kerdprasop, “Database reverses engineering based On Association rule mining”, IJCSI international Journal Of computer science, 2010.

30. Performance in Software Engineering. IEEE Trans. Softw. Eng., 2012; 38: 1276–1304. [CrossRef]
31. Radjenovic, D.; Hericko, M.; Torkar, R.; Zivkovic, A. Software fault prediction metrics: A Systematic literature
32. Report EBSE-2007-001; Keele University and Durham University: Staffordshire, UK, 2007. review. Inf. Softw. Technol, 2013; 55: 1397–1418. [CrossRef]
33. Ribu, Estimating, "Object oriented software projects With use cases", M. S. thesis, University of Oslo Department of informatics, 2001.
34. Runeson and Nyholm, "Detection of duplicate Defect report uses neural network processing", in Proceeding of the 29th international conference on Software engineering San Jose, CA, USA, November 2010; 3–4: 26–34.
35. Severity faults Software Engineering. IEEE Trans, 2006; 32: 771–789.
36. Shtern and Vassilios, "Review article advances in Software engineering clustering methodologies for software engineering", Tzerpos volume, 2012.
37. Sunita Tiwari and Neha Chaudhary, "Data mining and Warehousing" Dhanpati Rai and Co.(P) Ltd. First Edition, 2010.
38. Wen, S.; Li, Z.; Lin, Y.; Hu, C.; Huang, C. Systematic literature review of machine learning based software.
39. Zhou, Y.; Hareton, L. Empirical analysis of object-oriented design metrics for predicting high and low.