

Software Bug Prediction using Machine Learning Approach

Meenakshi, Dr. Satwinder Singh

M. Tech (Computer Science and Technology), Central University of Punjab, Bathinda.

Assistant Professor, Dept. of Computer Science and Technology, Central University of Punjab, Bathinda.

Abstract - Software bug prediction plays an important role in development and maintenance of good quality software. Bugs are some kind of code in any software which can cause failure. Major cost of any software project is used in maintenance of that software because of bugs. It is very important to predict bugs in earlier stage of a software development. So that the development team can identify them and resolve them which can improve software, quality, reliability, efficiency and reduces the cost of project. In this paper, ML is used to predict the bugs on the bases of historical data. At last we compare the results of two ML classifiers i.e. Naive Bayes and J48.

Key Words: Bug Prediction, quality, Machine learning, Naive Bayes, J48.

1. INTRODUCTION

Software bug prediction is defined as a process where we try to predict bug based on historical data. Software bugs can affect the software quality, reliability and maintenance cost. While developing software there are so many hidden bugs which can cause software failure in future. Around 40-60% of cost is used in maintenance of software so it is very to predict the bugs in earlier stages of software development. By predicting the bugs we can easily reduce the software failure rate. There are some techniques are available to predict the bugs. In this paper, Machine learning approach is used to predict the bugs in software on historical data. Naive Bayes and J48 Classifiers are used to predict the bugs.

2. LITERATURE REVIEW

There are many studies about software bug prediction using machine learning techniques. For example, In study [3] proposed machine learning approach. In which supervised learning is used to predict the bugs on the bases of historical data. There are three classifiers are used i.e. Naive Bayes, Decision Tree, Artificial Neural Networks. Results got from these three classifiers are compared with each other to find out the best model of machine learning to predict the bugs.

The study in [1] analysed that the software quality depends upon many factors like resources, executive team etc. Three types of machine learning approaches are used to predict the bugs i.e. based on classification method, based on clustering method and based on ensemble approaches.

The study in [4] a comprehensive study is done on different existing predictive models. A deterministic strategy should follow for predicting the bugs. A comparison is done between the models on the bases of recall and F-measure. By using good model in software development environment, will give a reasonably accurate prediction.

The study in [2] defines that the maintenance of software is challenging task because of unpredictable bugs and failures which tends to increase in software cost. Machine learning approach is used to identifying the best way of predicting those bugs on the bases of historical data. The different datasets are being compared with each other on the bases of f-measure, recall, precision. Decision tree classification provides better prediction rather than other models.

This paper gives a comparative study of Naive Bayes and J48. The supervised learning in machine learning approach is used to predict the bugs on the bases of f-measure, recall and precision. Bug prediction can improve the software success rate if prediction is used in earlier stages.

3. RESEARCH METHODOLOGY

In Research Methodology, the machine learning algorithm and the used tools are discussed. Also, the various performance measures are discussed which are used to analyse the experiment.

3.1 Data Collection

The datasets from promise data repository were used in the experiments. The datasets were collected from real software projects by NASA and have many software modules. Some of the product metrics that are included in the data set are, Halstead Content, Halstead Difficulty, Halstead Effort, Halstead Error Estimate, Halstead Length, Halstead Level, Halstead Programming Time and Halstead Volume, Cyclomatic Complexity and Design Complexity, Lines of Total Code, LOC Blank, Branch Count, LOC Comments, Number of Operands, Number of Unique Operands and Number of Unique Operators, and lastly Defect Metrics; Error Count, Error Density, Number of Defects (with severity and priority information).

	Kc1	Mc1	Ar1	Ar6	Mc2
Language	C++	C++	C	C	C
LOC	43k	63k	29k	29k	6k
Modules	2107	9466	121	101	161
Defect	325	68	9	15	52

Table 1: Dataset Information

3.2 Observation recorded

Accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

Precision is a measure of correctness and it is a ratio between correctly classified software bugs and actual number of software bugs assigned to their category. It is calculated by the equation below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall is a ratio between correctly classified software bugs and software bugs belonging to their category. It represents the machine learning method's ability of searching extension and is calculated by the following equation.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4. RESULTS AND DISCUSSIONS

The experiments performed were analysed for different parameters. These parameters help to infer various aspects of models.

4.1 Design Metrics Prediction Model

To predict the bugs based on metrics should be accurate as possible. The resultant metrics prediction model has following measures:

Supervised Learning						
Datasets	Kc1	Mc1	Ar1	Ar6	Mc2	Mean
Naïve Bayes	82.77	95.29	88.42	84.15	74.53	77.07
J48	84.54	99.42	90.08	83.16	70.80	86.27

Table 2: Performance based on Accuracy

Supervised Learning						
Datasets	Kc1	Mc1	Ar1	Ar6	Mc2	Mean
Naïve Bayes	0.81	0.96	0.89	0.82	0.71	0.77
J48	0.80	0.99	0.87	0.80	0.66	0.82

Table 3: Performance based on F-measure

Supervised Learning						
Datasets	Kc1	Mc1	Ar1	Ar6	Mc2	Mean
Naïve Bayes	0.83	0.95	0.88	0.84	0.74	0.76
J48	0.84	0.99	0.90	0.83	0.70	0.85

Table 4: Performance based on Recall

Supervised Learning						
Datasets	Kc1	Mc1	Ar1	Ar6	Mc2	Mean
Naïve Bayes	0.80	0.98	0.90	0.81	0.74	0.84
J48	0.80	0.99	0.85	0.78	0.68	0.82

Table 5: Performance based on Precision.

5. CONCLUSIONS

Various approaches have been proposed using different datasets, different metrics and different performance measures. Results reveal that the ML techniques are efficient approaches to predict the future software bugs. The comparison results showed that the Decision Tree classifier has the best results over the others. Moreover, experimental results showed that using ML approach provides a better performance for the prediction model than other approaches, such as linear AR model.

As a future work, we may involve other ML techniques and provide an extensive comparison among them. Furthermore, adding more software metrics in the learning process is one possible approach to increase the accuracy of the prediction model. Poisson regression model could be used and Neural network model to predict the open bugs count and its performance with other existing models could be used.

REFERENCES

- [1] Akmel, F., Birihanu, E., & Siraj, B. (2017). A literature review study of software defect prediction using machine learning techniques. *Int. J. Emerg. Res. Manag. Technol*, 6(6), 300-306.
- [2] Aleem, S., Capretz, L. F., & Ahmed, F. (2015). Comparative performance analysis of machine learning techniques for software bug detection. *ITCS, CST, JSE, SIP, ARIA, DMS*, 71-79.
- [3] Awni Hammouri, M. H. (2018). Software Bug Prediction using Machine Learning Approach. *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9 (no.2), 78-83.
- [4] N.Kalaivani, D. (2018). Overview of Software Defect Prediction using Machine Learning Algorithms. *International Journal of Pure and Applied Mathematics*, 118 (20), 3863-3873
- [5] Shruthi Puranika, P. D. (2016). A Novel Machine Learning Approach For Bug Prediction. *6th International Conference On Advances In Computing & Communications, ICACC*, 6 (8), 923-930.