Analysis of Different Machine Learning Techniques for Defect Prediction of a Software

Bhanu Pratap Rai, C. S. Raghuvanshi, Hari Om Sharan

Faculty of Engineering & Technology, Rama University, Mandhana, Kanpur, 209217, U.P., India.

bhanurai0@gmail.com;

Abstract

Every company requires reliable and effective software to fulfill its requirement, at a low cost. Defective software can be dangerous in all fields whether you are using it in development fields, medical fields, production fields, or any other. Prediction of defects at an early stage can help us to save money and provide reliable software but for prediction, it is mandatory to test the software and all its module, and it will take more than 50% cost of the total cost of the software. ML is very famous at this time and provides different algorithms for analysis and prediction like SVM, DT, RF, etc. This paper is basically an analysis of all machine-learning techniques from recent years which may provide helps to the researchers in SDP because still there is no proper tool or technique which can predict and remove the defect of software.

Keyword: *machine learning (ML), Software defect prediction (SDP)*

Introduction

It is very difficult process to develop quality software which can fulfill the requirement of user. This process requires combine effort of all software development teams for analyzing, planning, testing and execution. In testing phase we get most of the defects which reduces the quality of the software. According to the report [19], testing phase take more time and more than 50 % cost of the software, where finding and fixing the defect takes main place.

Another paper of Anon 2018a (Software fails watch) which was published in Tricentis, examined 606 famous software failures. This paper reported that 314 companies were involved in this analysis, assets worth $1.7 trillion, and 3.8 billion people were affected. Number of experiences have shown that the use of defective software have an adverse effect on the business. If we take an example of a defective life-critical system software of any hospital. It is very dangerous to use. The critical business applications need a reliable software, and it is the main challenge for the development team of software industry.

In modern time, software industry is growing and being more advanced. As we know that we are highly dependent of the software. These software must be defect free, effective, highly secure and reliable. It should have the capability of maintenance according to the requirement of the user and should be small in size. According to the definition of ANSI "The possibility that a piece of software will operate without errors for a predetermined amount of time in a predetermined environment is known as software reliability".

Software Defect Prediction (SDP) is mainly describe as a back-down from the requirement or specification of a software [1]. To improve quality of a software and reduce failures we can perform unit testing, code review or defect prediction etc. These activities are also known as quality assurance activities. Cost of these activities are approximately 80% of the total budget of a project [2]. If we want to reduce the cost we will have to find the defective modules first. For this, SDP has been introduced [3].If we are able to predict defect of software we can reduce software development cost and increase quality of the software. So With the help of SDP we can do following:

• Find the software bugs in early stage

• Allocate the test resources efficiently

• Minimize the development cost of software

• Increase the quality and productivity of the software

We realize that machine learning (ML) techniques are winning nowadays. ML is an application of artificial intelligence (AI) where we provide large dataset to the machine and train it with this dataset to predict the output according to the given input. As we know that SDP is very important .Figure-1 shows overview of SDP process. To predict the defects pre-processing of the dataset is very important to analysis defects and to identify defects. We can apply different algorithms of ML to train and test the model to predict defects.

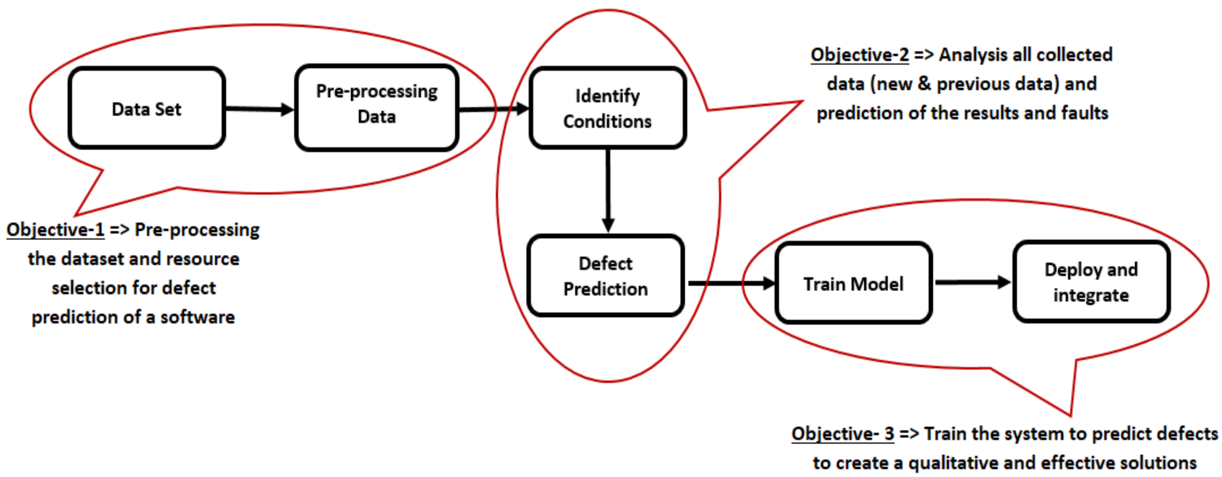

Figure 1: Overview of SDP Process

Related Work

Author of [4] discussed SDP strategies that have depend on software program metrics. Bagging, Support SVM, DT, and RF classifiers are recognized to carry out properly to expect defects. This paper studies and compares that supervised system getting to know and ensemble classifiers on 10 NASA datasets. The performances of different algorithms were evaluated using classification accuracy, F-measure, and ROC-AUC metrics.

Author of [5] compared supervised machine learning algorithms and group classifiers on 10 data-sets of NASA. The results of these algorithms were calculated using ROC-AUC metrics, classification accuracy, and F-Measure. The results of the experiment showed that bagging with DS, RF, and AdaBoost where RF performed well.

Researchers in [6] used NASA datasets for SDP with several classification techniques of ML. SVM, DT, RF ,Naive Bayes (NB), Radial Basis Function (RBF), Multi-Layer Perceptron (MLP), K Nearest Neighbor(KNN), kStar (K*),One Rule (OneR), and the performance of these techniques are calculated by different measures such as: F-Measure, Precision, Accuracy, Recall, MCC, and ROC area. This result can be used as baseline for other researches.

In [7], authors propose novel methodologies, CLAMI and CLA that show the potential for imperfection expectation on unlabeled datasets in an automated manner without need for manual exertion. The key thought of the CLA and CLAMI approaches is to mark an unlabeled dataset by utilizing the magnitude of metric value. In our observational concentrate on seven open-source projects, the CLAMI approach drove to the promising expectation exhibitions, 0.636 and 0.723 in normal f-measure and AUC that are similar to those of defect prediction in view of supervised learning.

This study [8] compare the ability of different feature extraction technique. The study of [8] SDP dataset of NASA is used and in this dataset 'min max normalization' is used for data preprocessing. To check the performance of the technique ROC-AUC and accuracy are used as performance measure.

Literature Outcome

For SDP we will have to follow several steps. Figure-2 shows the general life cycle of SDP [9] where we test the module of software and check this module is defective or not, if it is defective then try to fix the defect.

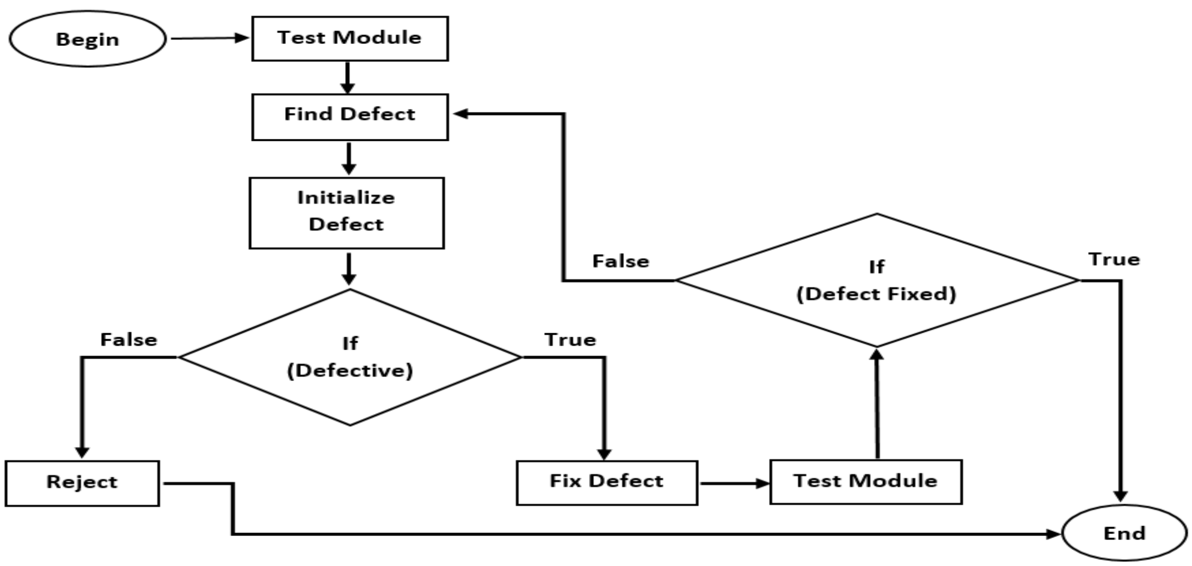

Figure 2: Life Cycle of Software Defect Prediction

In this study, several papers were analyzed from the year 2018 which may help researchers in analysis and prediction. Figure-2 shows different tools and techniques of AI which can be used in SDP. In this figure, we can see all ML techniques supervised and unsupervised, performance measurement techniques, and deep learning techniques. For the analysis, we need a dataset that may be public or custom to prepare training and testing.

Approaches of AI need historical data to predict. We will have to install additional software for this. The cost of this process is very high but it has a wide possibility to find the defect. If we compare it with the manual testing approach, the cost of the manual approach is very low. Manual testing did not need historical data and takes lots of time to execute. It does not require any additional software to install but limited probability to find defects.

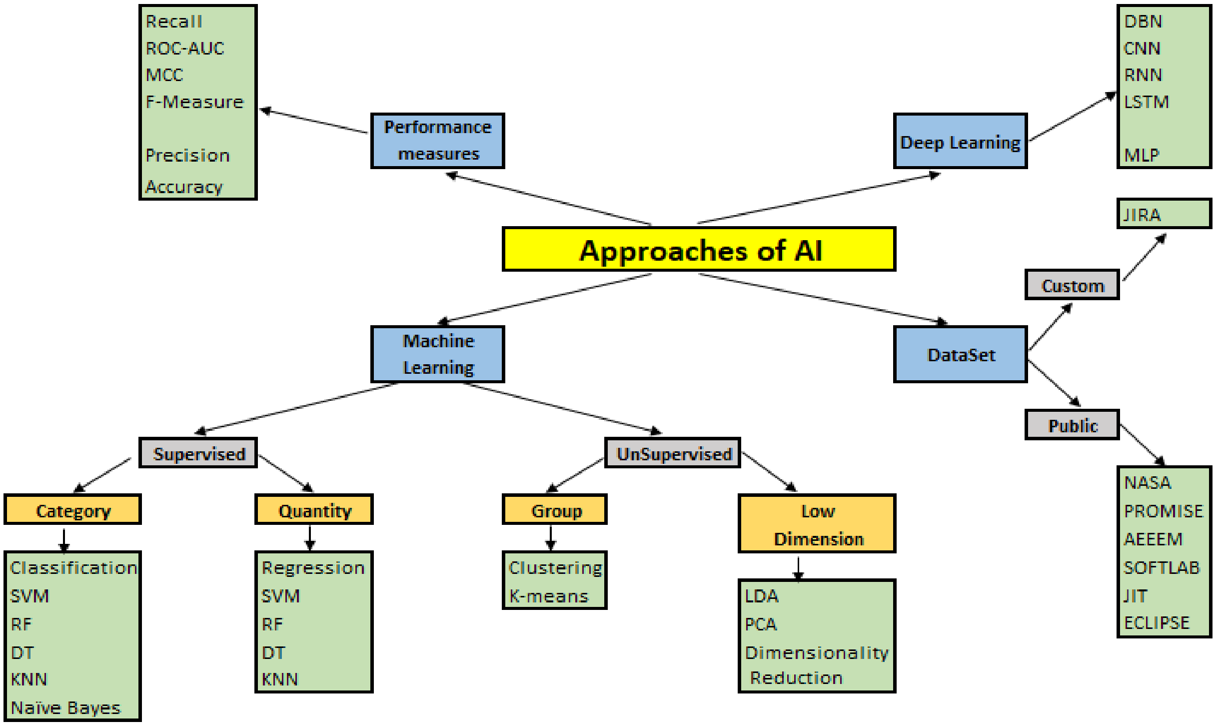

Figure 3: Tools and Techniques of AI for SDP

Every researcher needs to know the different defect-finding approaches, the available dataset, what are the different frameworks/tools available for SDP. Table-1 is the answer to these key questions. Table-1 shows all the analyses of the papers from the year 2018. In this table, we can see the used algorithms in the paper, what dataset is used in this paper, and in which year it is published. We can also see the result of the paper which may help researchers to analyze and select which algorithm is suitable for the SDP process.

In Year	Dataset	Algorithm	Best performance result	Paper
2022	PROMISE	PCA, LDA, K-PCA , Auto-encoders (AE) with SVM	To reduce the dimensionality of SDP, AE perform well	Bhanu P Rai et al. (2022)
2020	NASA	SVM, MLP, NB, RF, J48, KNN	RF perform well on the PC2 dataset with result 0.99 value	Naseem et al. [10]
	NASA	SVM, DT, KNN ,RBF , MLP , Ensemble learning, NB, RF	With Accuracy = .87, 'F-Measure = 0.75, MCC = 0.669'. Ensemble approach perform well	Ali et al. [11]
	PROMISE	KNN (Bagging based)	With result AUC=0.726 for the CM1 dataset. KNN with Bagging with Feature selection using PCA performs the best	Saifan and Abu-wardih [12]

2019	NASA	MLP,RF, KNN,	According to the result with 'R square value = 0.9969'. KNN perform well	Abdul shaheed et al. [13]
	PROMISE	WNB-ID (Weighted Naive Bayes)	With the value of F-Measure = 0.8669 , WNB-ID performance was good for the dataset POI-2.5.	Ji et al. [14]
2018	NASA	TSWNB (Two Stage Naive Bayes)	With the result value of AUC =0.7835 for the PC4 dataset. TSWNB perform well	Ji et al. [15]
	NASA	SVM , NB , RF, RPart	In SDP , Ensembles classifiers with decision-making strategies perform well	Bowes et al. [16]
	PROMISE	BoostingJ48, , LR, Boosting Naive CODEP (Logistic) Max, RF	BoostingJ48, RF, and Max have good F-scores in comparison of the F-measure of the CODEP(Logistic) by 0.32%, 2.32%, and 36.87%, respectively.	Zhang et al. [17]

Table 1: Analysis of Machine Learning Techniques

Conclusion

Every company requires reliable and effective software to fulfill its requirement, at a low cost. Defective software can be dangerous in all fields whether you are using it in development fields, medical fields, production fields, or any other. Manual testing takes a lot of time to test and predict the SDP. Approaches of AI are very effective in SDP. It is a costly process and requires additional software to install but it has a wide possibility to find the defect. ML is a subset of AI. This paper will help researchers to compare ML algorithms.

References

[1] Norman E Fenton and Martin Neil. A critique of software defect prediction models. IEEE Transactions on software engineering, 25(5):675–689, 1999.

[2] Strategic Planning. The economic impacts of inadequate infrastructure for software testing. National Institute of Standards and Technology, page 1, 2002.

[3] Jaechang Nam, Wei Fu, Sunghun Kim, Tim Menzies, and Lin Tan. Heterogeneous defect prediction. IEEE Transactions on Software Engineering, 44(9):874–896, 2017.

[4] Mahesh Kumar Thota, Francis H Shajin, P Rajesh, et al. Survey on software defect prediction techniques. International Journal of Applied Science and Engineering, 17(4):331–344, 2020.

[5] Abdullah Alsaeedi and Mohammad Zubair Khan. Software defect prediction using supervised machine learning and ensemble techniques: a comparative study. Journal of Software Engineering and Applications, 12 (5):85–100, 2019.

[6] Ahmed Iqbal, Shabib Aftab, Umair Ali, Zahid Nawaz, Laraib Sana, Munir Ahmad, and Arif Husen. Performance analysis of machine learning techniques on software defect prediction using nasa datasets. International Journal of Advanced Computer Science and Applications, 10(5), 2019.

[7] Jaechang Nam and Sunghun Kim. Clami: Defect prediction on unlabeled datasets (t). pages 452–463, 2015

[8] Malhotra R, Khan K. A study on software defect prediction using feature extraction techniques. In2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) 2020 Jun 4 (pp. 1139-1144). IEEE.

[9] Pachouly J, Ahirrao S, Kotecha K, Selvachandran G, Abraham A. A systematic literature review on software defect prediction using artificial intelligence: Datasets, Data Validation Methods, Approaches, and Tools. Engineering Applications of Artificial Intelligence. 2022 May 1;111:104773.

[10] Naseem, Rashid, et al., 2020. Investigating tree family machine learning techniques for a predictive system to unveil software defects. Complexity 2020.

[11] Ali, Umair, et al., 2020. Software defect prediction using variant based ensemble learning and feature selection techniques. Int. J. Modern Educ. Comput. Sci. 12 (5).

[12] Saifan, Ahmad A., Abu-wardih, Lina, 2020. Software defect prediction based on feature subset selection and ensemble classification. ECTI Trans. Comput. Inf. Technol. (ECTI-CIT) 14 (2), 213–228.

[13] Abdul shaheed, Mohamed, et al., 2019a. Mining historical software testing outcomes to predict future results. Compusoft 8 (12), 3525–3529

[14] Ji, Haijin, et al., 2019. A new weighted naive Bayes method based on information diffusion for software defect prediction. Softw. Qual. J. 27 (3), 923–968.

[15] Ji, Haijin, et al., 2018. A two-stage feature weighting method for naive Bayes and its application in software defect prediction. Int. J. Perform. Eng. 14 (7), 1468.

[16] Bowes, David, Hall, Tracy, Petrić, Jean, 2018. Software defect prediction: do different classifiers find the same defects? Softw. Qual. J. 26 (2), 525–552

[17] Zhang, Zhi-Wu, Jing, Xiao-Yuan, Wu, Fei, 2018a. Low-rank representation for semi-supervised software defect prediction. IET Softw. 12 (6), 527–535.

[18] Bhanu P Rai, Dr C. S. Raghuvanshi , Prediction of Software Defect using Feature Extraction Technique: A Study. In2022 4th International Conference on Artificial Intelligence & Robotics in Life Science ICAR-2022

[19] Anon, 2018b. The cost of poor-quality software in the us: a 2018 report (2018, september 26). Retrieved from: https://www.it-cisq.org/the-cost-of-poor-qualitysoftware-in-the-us-a-2018-report/The-Cost-of-Poor-Quality-Software-in-the-US2018-Report.pdf.