International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Prediction of Software Design Defect using Enhanced Machine Learning Techniques

Karthikeyan C, Makineni Vinay Chandra, Jaswanth Santhosh Nadh, Mellempudi Nikitha

Abstract—Prediction of software detection is most widely used in many software projects and this will improve the software quality, reducing the cost of the software project. It is very important for the developers to check every package and code files within the project. There are two classifiers that are present in the Software Package Defect (SPD) prediction that can be divided as Defect–prone and not-defect-prone modules. In this paper, the merging of Cost-Sensitive Variance Score (CSVS), Cost-Sensitive craniologist Score (CSLS) and Cost-Sensitive Constraint Score (CSCS). The comparitive analysis can be shown in between the three algorithms and also individually.

Keywords: Software prediction, CSLS, CSCS, CSVS.

I. INTRODUCTION

Presently every software development is done in many organizations to overcome issues in the manual process. It is very important to find the accurate and exact software development defects. The identification of software defects in any software helps to improve the quality of the software. To calculate the software quality the design or coding is used to find the errors or exceptions in the program [1]. This is called a defect in software program. Most of the software package defect prediction studies have used machine learning techniques [11]. the primary step to make a prediction model is to get instances from software package archives like version management systems, issue trailing systems, e-mail archives, and so on. every instance will represent a system, a software package part (or package), a ASCII text file, a class, a operate (or method), and/or a code modification in keeping with prediction graininess. once generating instances with metrics and labels, we are able to apply pre-processing techniques, that are common in machine learning. Pre-processing techniques employed in defect prediction studies embody Feature choice, Dimension reduction, Classification, Prediction and eventually Performance analysis [2].

Manuscript published on January 30, 2020.
* Correspondence Author
Karthikeyan C*, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.
Makineni Vinay Chandra, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.
Jaswanth Santhosh Nadh, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.
Mellempudi Nikitha, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.

The flow chart below depicts the whole method of software package defect prediction. The historical knowledge, as well as varied software package parameters is captured from software systems. With the ultimate set of coaching instances, we are able to train a prediction model. The prediction model will predict whether or not a replacement instance is defect prone or not-defect-pron
Software defect prediction is the e.

II. LITERATURE SURVEY

Most popular research in finding the defects in software projects by using the parameters and machine learning (ML) methods. The author [3,10] focuses on identifying the total number of defects that reduce the time and cost of software development. The author used the ID3 algorithm to detect classification. This algorithm generates the decision tree from a dataset. The defects classification is done on five parameters such as length of program, volume, difficulty, processing time and effort.
Before building a forecast model, we have a tendency to tend to would possibly apply the ensuing procedures: embody call, institutionalization, and commotion taking care of. With the pre-processing procedures anticipated, forecast execution probably may be improved within the associated investigations. Analysts even have anticipated methodologies for cross-venture deformity forecast. Most delegate contemplates portraying above square measure directed and confirmed beneath the within forecast setting, as an example expectation models were planned and tried within the same endeavour. In any case, it's intense for contemporary out of the box new comes, that require additional improvement recorded info, to create various prediction models [4]. Various methods alter associate expectation model by selecting comparative cases, renovating information esteems, or increase a contemporary out of the plastic new model [5].

III. METHODOLOGIES

In this paper we have used the following methodologies to select the enhanced algorithms and measure the cost sensitive learning to predict the accuracy of the software by using machine learning algorithms.

Methodologies:

- ►Enhanced Feature Selection (EFS)
- ►Cost Sensitive Learning

Machine Learning Algorithms:

- ► Naive Bayes Classifier
- ► Svm
- ► J48 Decision Tree
- ► Rule Based Classification
- ► R Part

A. Enhanced Feature Selection (EFS)

Previously from the past, many year's lots of research is done on EFS with the various algorithms based on pattern recognition and ML algorithms. The main aim of the feature extraction is to extract the features in various software defect predictions [9].
The options are initial extracted on the premise of the three algorithms. The options so obtained are then combined along and given as input to the projected rule to get a replacement set of features.
Variance score (VS) could be a simple unattended analysis measure of many alternatives. Laplacian Score (LS) not exclusively lean towards alternatives with bigger variances that have extra agent control, anyway conjointly favours choices neighbors could be a semi-administered highlight decision technique, that performs include decision in step with the requirement defensive capacity of choices Any place choices which will best save the must-connect imperatives [6].
The options are initial extracted on the premise of the three algorithms. The options so obtained are then combined along and given as input to the projected rule to get a replacement set of features [8].

B. Cost Sensitive Learning

The live of performance of a machine learning formula is predicated on its accuracy of classifying a knowledge set. Here, we have a tendency to decision the category from the first class to the (c-1) the class the in-group class, whereas the c^{th} category is named the out-group class. Then, we are able to categorise misclassification prices into 3 types:
1) The false acceptance cost is i.e., this is the misclassifying cost with the sample that is from outside the group and it is inside the group class.
2) The false rejection cost is to misclassify the sample from the inside group is considered as outside the group.
3) False identification cost is to misclassify the sample from one inside group as being from another outside group class.

IV. OBJECTIVE PREDICTION

The features extracted mistreatment increased feature choice are utilized in the check knowledge and checked if the worth falls at intervals the vary [7]. The main objective is to seek out out sheep the given software system is defected from the info they need given or not and to perform analysis to urge the simplest formula to resolve the defected software [8].
The results of prediction will be expressed as a confusion matrix show below.
True positive (TP): this is called as defect prone.
False positives (FP): this is not-defect-prone module.
True negative (TN): this is not-defect-prone module.
False negative (FN): this is defect prone module foreseen as not-defect-prone.

V. PERFOMANCE ANALYSIS

The performance of the proposed system is calculated based on the learning of cost-sensitive eventualities, sum of misclassification cost, and the classification results are calculated as follows

A. False Positive Rate (FPR)

The percentage of cases where a data was classified to normal data, but in fact it did not.

$$FPR = \frac{FP}{FP + TN}$$

B. False Negative Rate (FNR)

The percentage of cases where an data was classified to abnormal data, but in fact it did.

C. Evolution Results

Iris data set performance with naive bayes algorithm.

$$FNR = \frac{FN}{FN + TN}$$

D. Sensitivity

The original positives which are accurately find the calculation of the sensitivity. It relates to the ability of the test to identify positive results.

$$Sesitivity = \frac{No.ofTP}{No.ofTP + No.ofTN}$$

E. Specificity

The real negatives which are accurately find the calculation of the specificity. It relates to the ability of the test to identify negative results.

$$Specificity = \frac{No.ofTN}{No.ofTN + No.ofFP}$$

F. Accuracy

This will calculate the overall accuracy of the clusters.

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

VI. RESULTS

Here we've used jdk ,weka and jfree_chart libraries to run the java program and also the wood hen tool is employed to modify the given information sets with the on the market algorithmic rules to seek out the utmost accuracy we are able to discover and check sheep at that algorithm it's less defects then implement them therewith algorithm.

Naive Bayes Classifier

	Class		
Attribute	Iris-setosa	Iris-versicolor	Iris-virginica
	(0.33)	(0.33)	(0.33)

Sepal length			
mean	4.9913	5.9379	6.5795
std. dev.	0.355	0.5042	0.6353
weight sum	50	50	50
precision	0.1059	0.1059	0.1059

Capabilities: [Nominal attributes Binary attributes, Unary ttributes, Empty nominal attributes, Numeric attributes, Missing values, Nominal class, Binary class, Missing class values]
Dependencies: []
min # Instance: 0
KAppa Statistics:: 0.94

Sepal width			
mean	3.4015	2.7687	2.9629
std. dev.	0.3925	0.3038	0.3088
weight sum	50	50	50
precision	0.1091	0.1091	0.1091

Petal length			
mean	1.4694	4.2452	5.5516
std. dev.	0.1782	0.4712	0.5529
weight sum	50	50	50
precision	0.1405	0.1405	0.1405

Petal Width			
mean	0.2743	1.3097	2.0343
std. dev.	0.1096	0.1915	0.2646
weight sum	50	50	50
precision	0.1143	0.1143	0.1143

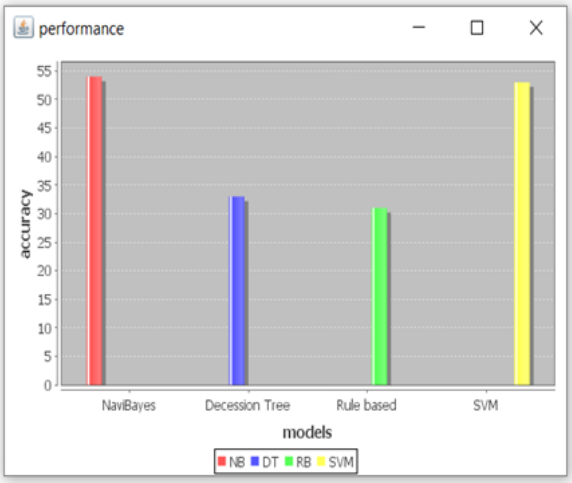

Figure: 1 Performance of Algorithms

-Measure:: 0.9599839935974391
Confusion MAtrix:: Confusion matrix:
a b c <-- classified as
50 0 0 | a = Iris-setosa
0 48 2 | b = Iris-versicolor
0 4 46 | c = Iris-virginica
Capabilities: [Nominal attributes, Binary attributes, Unary attributes, Empty nominal attributes, Numeric attributes, Missing values, Nominal class, Binary class, Missing class values]
Dependencies: []
min # Instance: 0
KAppa Statistics:: 0.94
F-Measure:: 0.9599839935974391

Confusion Matrix::Confusion matrix:

a b c <-- classified as
50 0 0 | a = Iris-setosa
0 48 2 | b = Iris-versicolor
0 4 46 | c = Iris-virginica

Mean Absolute Error	0.03422
RMSE	0.155
Relative Absolute Error	7.6997

Root Relative Squared Error	32.8794
fMeasure	0.95998
Naive Bayes Classifier	
Mean Absolute Error	0.17608
RMSE	0.41388
Relative Absolute Error	67.304
Root Relative Squared Error	114.49
fMeasure	0.81959

From the outputs we have got from the promise software data sets we have calculated the accuracy of the respected data sets with different algorithms and obtained a comparative results from that the best accuracy algorithm is taken as it shows more accuracy towards TP, TN, FP, FN. The given data sets they say that the software is error free and there are no defects but by running them through different algorithms we get actual phenomenal outputs with change in the TP,TN,FP,FN. From this analysis we can say for the respected data set we can implement it with the respected algorithm which has highest accuracy can represent the data set with lowest defect and can run most efficiently by implementing those errors with that algorithm.

VII. CONCLUSION

The mean accuracy is calculated with the naive bayes classifier and the result of the dataset is 83.47. The performance is more on CM1, PC1 and for this datasets the accuracy is 95%. The low performance is observed with the dataset KC2 and for this the accuracy is 50%. The rule based is most widely performed well on CM1 and PC1 and total accuracy for various datasets is 89.14%. The performance of the SVM is done by using various machine learning methods to improve the overall accuracy upto 89%. Thus it is known that the proposed system shows the performance is 98%.

REFERENCES

1. M. Liu, L. Miao and D. Zhang, "Two-Stage Cost-Sensitive Learning for Software Defect Prediction," in IEEE Transactions on Reliability, vol. 63, no. 2, pp. 676-686, June 2014.
2. Qin Z., Zhang C., Wang T., Zhang S. (2010) Cost Sensitive Classification in Data Mining. In: Cao L., Feng Y., Zhong J. (eds) Advanced Data Mining and Applications. ADMA 2010. Lecture Notes in Computer Science, vol 6440. Springer, Berlin, Heidelberg.
3. Dan Sun and Daoqiang Zhang. 2010. “Bagging Constraint Score for feature selection with pair wise constraints. Pattern Recognition”. 43, 6-June- 2010, 2106-2118.
4. C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse and A. Napolitano, "A Comparative Study of Data Sampling and Cost Sensitive Learning," 2008 IEEE ICDM Workshops, Pisa, 2008, pp. 46-52.
5. Zhi-Hua Zhou and Xu-Ying Liu. 2006. “On multi-class cost-sensitive learning”. In Proceedings of the 21st national conference on AAAI'06, Anthony Cohn (Ed.), Vol. 1. AAAI Press 567-572.
6. Abraham Bernstein, Jayalath Ekanayake, and Martin Pinzger. 2007. “Improving defect prediction using temporal features and non linear models”. IWPSE '07. ACM, New York, NY, USA, 11-18.
7. Y. Bo and L. Xiang, “A study on software reliability prediction based on support vector machines,” in Proc. IEEE Int. Conf. Ind. Eng. Eng.Manag., Singapore, 2007, pp. 1176–1180.
8. L. Guo, Y. Ma, B. Cukic, and H. Singh, “Robust prediction of defect proneness by random forests,” in Proc. 15th Int. Symp. Software Rel.Eng., 2010.
9. Mohiddin, S. K., Kumar, P. S., Sai, S. A. M., & Santhi, M. V. B. T. (2019). Machine learning techniques to improve the results of student performance. International Journal of Innovative Technology and Exploring Engineering, 8(5), 590-594.
10. Rajesh, N., Maneesha, T., Hafeez, S., & Krishna, H. (2018). Prediction of heart disease using machine learning algorithms. International Journal of Engineering and Technology(UAE), 7(2.32 Special Issue 32), 363-366.
11. Sagar, P. V., Moparthi, N. R., & Mandhala, V. N. (2018). Probabilistic estimation of software development effort techniques using machine learning. International Journal of Engineering and Technology(UAE), 7, 1085-1090.

AUTHORS PROFILE

Dr. C. Karthikeyan, having more than 18 years of teaching experience in Computer Science Engineering background in India and Abroad.He has been awarded Ph.D. in Computer Science & Engineering from Jawaharlal Nehru Technological University (JNTU) Hyderabad. He has published more than 30 papers in various International / National Journals, Conferences and book chapters. He organized various Seminars, workshops, symposium and conferences in India and abroad. He is having Professional membership in IEEE, IACSIT, ACEEE, ISTE, SDIWC and IAENG Associations.

Vinay Chandra Makineni, is pursuing Bachelor’s Degree in Computer Science at KL Educational Foundation and about to receive his degree by 2020-May. He was interested in pursuing his career in Machine Learning and Artificial Intelligence. He has done an internship in Software engineering and involved in a project “Linux Programming Architecture” during this period in internship in Swecha. He had developed and designed the project with the help of my colleagues. Hehas done projects in Platform-Based development, Object Oriented Programming, Machine Learning and Data Analytics during his years of study. He had done certifications in Natural Language Processing and in Ionic and Angular JS. Hewas planning to pursue Masters in Artificial Intelligence abroad.

Jaswanth Santhosh Nadh Neelam, pursuing final year B. Tech in KLEF Deemed to be university, will receive my B. Tech Degree by MAY-2020. He had done my Internship in Aspirevison Tech and completed my project on Computer Science background that includes Mobile Operator Portal Using OOPS (JAVA) , Apache Tomcat as a backend server and He has helped in one of the algorithm in the project. His research interest is Machine Learning, Internet of Things. He also participated in NSS Activities organized by CEA (Center for Extension Activities) and member of Youth Red Cross.

Nikitha Mellempudi, studying final year B.tech in KLEF Deemed to be University, and will receive B.tech degree by May 2020. She had done my internship in BSNL networks on may 2018. She had worked on different projects on Computer Science during my Under Graduation that includes Food Delivery App. She have helped in one of the algorithm in the project.. Her Research Interests are Image processing, Machine Learning and IOT. She had done a project “Skin Disease Detection” based on Machine Learning in my 3rd year. Also Participated in NSS Activities organized by CEA (Center for Extension Activities).