IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.7, Special Issue.1, Jan 2019 E-ISSN: 2347-2693

A Framework of Software Defect Prediction By Data Mining Techniques Using Historical Data Set and Intelligent Agents

Amitava Bondyopadhyay

Department of Computer Science, Mankar College, Burdwan, India

*Corresponding Author: in.amitava@gmail.com

Available online at: www.ijcseonline.org

***Abstract*—** Defect prediction for a software system is a technique that is used extensively nowadays to predict defects from historical database. But only a good data mining model is not enough to extract defect from software bug record. Intelligent agents are helpful in this case by making the decision process easier at some point. This paper describes frame work to generate software defect from the historical database and also propose one algorithm that is used find policy to forecast software defects efficiently than the current methods.

***Keywords*—** Cost, Classification, Intelligent agents ,Data mining, Database, Defect, Testing

I. INTRODUCTION

The main intend of software development process is to build high-quality software carefully. Getting a good quality software is the need of the hour. For this reason I need to reduce cost and improving the overall working of the testing process . Hence measuring software defects at early stage is extremely important. So if I guess early defects of software , then it will be helpful in increasing the software quality by minimizing the errors during the maintenance phases. Hence our aim should be to find an efficient method that can be achieved from getting knowledge from the previous mistake and making a newer and more correct system . In today's world various data sets are accessible which can be used in order to get clues regarding possible defects that may remain in the software system.

From early days data mining techniques are applied in constructing software fault prediction for improving the software quality. But that is not enough in some decision making process and hence the concept of intelligent agent comes in this regard which helps the mining algorithm in finding the right decision while splitting the data into training and test data. Also I need to identify high risk modules (having high number of faults) at the earliest which can be helpful in quality enhancement effort.

II. SOFTWARE DEFECT PREDICTION

A software defect [5] is a mistake or fault that team always want to produce a quality software with minimum defects. To increase the software quality, high risk components from the software project should be removed as soon as possible. Software defects always incur cost in terms of quality and time**.** To identify and rectify defects is important in a software system which may create a wrong or unpredicted result, or prevent the software from getting a good quality . It is not possible to eliminate each and every defect in one software but it can be minimized and their adverse effects can be reduced.

By defect predictor, I mean, a technique that guides testing activities in software development lifecycle. According to Brooks[6], testing phase accounts for half of the total effort. Harold and Tahat[2] also at one with the view of Brooks. Hence the duty is mainly on the tester and they have to find where the defects might exist before they start testing.

It will help them to assign their limited resources in an effective way. One of the key use of the defect predictor is to create an order which is to be verified and validated by a team of experts. Defect predictors are also useful in finding defects efficiently in lesser amount of time. This process is very helpful, because it gives prior warning of what modules need modification, and giving them ample time to finish the rework prior to schedule.

III. ISSUES WITH THE EARLIER WORK

As per the current research all previously software prediction model can be useful if enough amounts of data is available to feed the model. Here for the lack of good data mining model, taking out of defects from large software bug repository is a very tiresome process and the desired result is not always possible. Existing prediction models[3] that are created previously using sampling and training dataset fails to give

the exact details of number of fault in the software and circulation of error among modules of the software system . It also makes the fit database imbalanced(for highly twisted data set) and thereby making them unsuitable for prediction purpose. Sometimes the results of more balanced dataset are also unsatisfactory .In these cases early life cycle data are not giving the sufficient clue for finding the fault prone models. One can not be able to find good result if different software repositories are mined. Single classifier or combining different classifier would not yield any good result. Supervised learning are also not suitable for high level modules although they can predict good result at same logical levels. The current lot of classifier based model are not always giving good results. Because at every case the decision making process is not fit enough to take accurate decision .like while splitting the historical data into training dada and test data ,there is a need of good decision making process. Also while comparing the result the task agents are helpful in taking decision.

VI.ACCURATE DEFECT PREDICTION MODEL

The need of the hour is a correct defect prediction framework for large software system which are more prone to error. Traditional decision tree are currently used in classification[4] for forecasting of defective and non-defective part of a software system.. However they have some drawbacks. Hence the need of a suitable mining model can improve the whole existing process.

V. INTELLIGENT AGENT CONCEPT

Intelligent agent thought and multi-agent systems stand for a new way of making many decision jobs. By intelligent agents I mean independent software entities that can navigate diverse computing environments . It has the ability to work alone or with other agents, for obtaining user-defined goals. Now a days agents can be used in various decision making process. Here I propose the agent-based technology to achieve the decision making of which proportion the historical data to be split into training and test data to make a great prediction model that can achieve a creditable defect prediction and to achieve considerably superior decision process that will help to create a excellent quality design .

VI. PROPOSED FAULT PREDICTION FRAMEWORK WITH THE HELP OF INTELLIGENT AGENTS

The most important thing I need to keep in our mind while building a defect prediction model is to decide on the best learning scheme or algorithm from where I can achieve our goal . So that by this performance, learning schemes can be made for future data. Also one problem arises on how to split the historical data into training data and test data. Here I propose to take the help of intelligent agent concept. As per problem it will help to take the decision in this regard. I brake the framework into three parts , Splitting the historical data into training data and test data by intelligent agent, Choosing of best learning algorithm and at last predicting defect. At the Learning method estimation stage, the proposed algorithm will bring the best from a set of contending method by making assessment of all these using past data and with the help of task agent. In our framework **,** I can see whole the historical data are broken into training data and test data by the help of intelligent agent and the test data are never used to build the learners. Intelligent agents are very much helpful in breaking the training data and test data into right proportion . Hence by this way I can select the best schemes among all given schemes by comparing them with the help of task agents. Once the performance report is available I can proceed to the defect prediction stage and thereby I can easily build the desired prediction model.

VII. PROPOSED FRAMEWORK

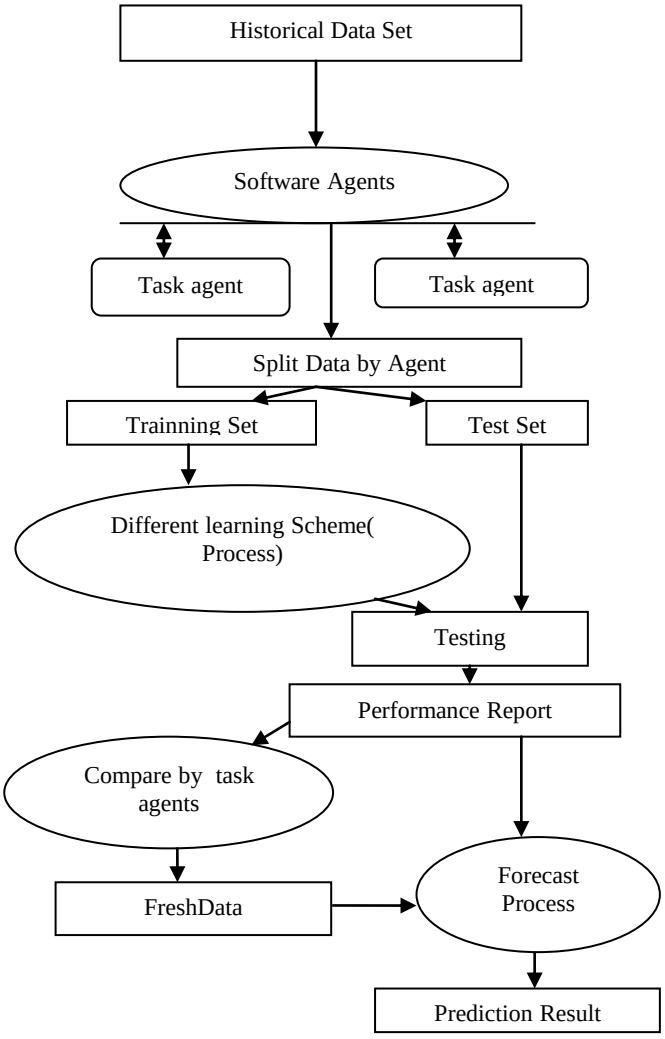

Figure 1

VIII. LEARNING METHOD ESTIMATION

Choosing the best learning algorithm is a major thing of the software defect prediction framework. The main problem I face is that the process of dividing the historical data into training data and test data. And our way of resolving it is already discussed ie, by the help of intelligent agent and task agent. From figure 1 we can see that they are taking the decision of which proportion the data is broken into two part to get the optimal result. Here I should keep in mind that the test data should be independent while constructing the learner. This should be treated as the required precondition. I also used several repetition of actual method to make an accurate predictive model as per the need of the problem which will be best suited for a particular case. Then the validation results are averaged over several rounds. At last the training data and best learning method are used to make a learner which help in finding the desired learning algorithm.

IX. PROPOSED ALGORITHM

Learning_Method_Estimation_Algoritm

Step 1: Input the historical Data and , different competing learning schemes
Step 2: Input how many no of data set is given , name it " SET" ; Input How many times each data set will be repeated; name it "REP"
Step 3: initialize x=0;
Step 4: Repeat step5 through step 15 until x<SET
Step 5: initialize y=0;
Step 6: Repeat step7 through step 14 untill y<REP
Step 7: Create an array Data , initialize it with whole lot of historical Data and then generate REP no of bins from Data
Step 8: Read data set one by one from array data
Step 9: Obtain Train_data=Agent_decision(data)
Step10: Train[x]= Train_data //Training Data percentage calculated by agents from historical data;
Step11:(learner,bestAttrs)=Learning_method (train, learning scheme)
Step 12: Test Data=Data[x]-Train[x]; Keep remaining Attrs for testing purpose other than the train data
Step 13 : Result=TestClassifier(Test[x],Learner);
Step 14: y++; goto step 6
Step 15: x++; goto step 4
Step 16: Compare result by the help of task agents
Step 17: Average _outcome=(1/SET*REP)∑Result

X. PREDICTING DEFECTS BY THE PROPOSED METHOD

The prediction of defect by our method is quite easy. First I construct the predictor, and then the defects are predicted. In both the cases I take the help of intelligent agents. A learning scheme is found during the prediction construction and that is too comparing the Performance Report with the help of task agents. Next I get the selected learning scheme .With the help of that I can evaluate a learning scheme for creating the learner . After that the average of all rounds are calculated and thereby final performance is achieved. This shows that the evaluation is actually covering all the data. Hence, our aim should be to use all of the previous data to make the predictor, and I can expect that the predictor which is constructed by this method has a stronger generalization sense. For that reason I applied the intelligent agent concepts in decision making process. All the new data that are coming next are preprocessed in exactly the same manner as I did for the historical data. At last the predictor, which I made, can be used to predict software defect with the whole lot of new data.

XI. BENEFIT OF THE PROPOSED METHOD

The proposed method works significantly well comparing the other similar methods. To work with it we do not need large amount of data which is often not available. . One can be able to find good result if different software repositories are mined. Single classifier or combining different classifier would yield slightly better result. Because at every case the use of intelligent agent, the decision making process becomes fit enough to take accurate decision.

CONCLUSIONS

In this paper, I tried to make a standard structure for software defect prediction. It involves assessment and prediction. In the assessment stage, I ensure that different learning schemes are examined and the best one is getting selected. Our main achievement here is to introduce the concept of intelligent agent that that takes various decision of this defect prediction model. Here prediction stage ensures making of best learning scheme to build a predictor and that too with the help of task agents. So all the previous data and the predictor can be helpful in predicting error on any recently created data.

REFERENCES

[1]. Ms. P.J Kaur, Ms. Pallavi, "Data Mining Techniques for Software Defect Prediction" , International Journal of Software and Ib Sciences (IJSWS)
[2]. W Sunindyo, T Moser, D Winkler , "Improving Open Source Software Process Quality based on Defect Data Mining" , Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems,Vienna University of Technology.
[3]. K.B.S Sastry, Dr.B.V.Subba Rao, Dr K.V.Sambasiva Rao, "Software Defect Prediction from Historical Data" ,International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2013
[4]. N Azeem, S Usmani, "Analysis of Data Mining Based Software Defect Prediction Techniques",Global Journal of Computer Science and Technology, Volume 11 Issue 16 Version 1.0 September 2011

[5]. T Xie, S Thummalapenta, D Lo, and C Liu, "Data mining for software engineering." Computer, 42(8):55-62, 2009.
[6]. M Baojun, K Dejaeger, J Vanthienen, and B Baesens, "Software defect prediction based on association rule classification" SSRN 1785381, 2011.

Authors Profile

Mr. A.Bondyopadhyay pursed Master of Computer Application from University of Burdwan, Burdwan in 2006 and was the Gold medalist in the same. He also did M.Phil in Computer Science from Annamalai University in year 2009. He is currently pursuing Ph.D. from the University of Biurdwan and currently working as Assistant Professor in Department of Computer Sciences, Mankar College since April,2010. He has published more than 05 research papers in reputed journals and conferences. His main research work focuses on Software Quality, Data mining, Defect data, Intelligent Agents . He has more than 12 years of teaching experience and 5 years of Research Experience.He has also successfully completed one UGC Minor Research Project.

