Software Fault Prediction Exploration Using Machine Learning Techniques

P. Patchaiammal, R. Thirumalaiselvi

Abstract: ***Fault prediction is one of the major activity of quality assurance. Fault prediction plays a significant role in the reduction of software cost and time. Even though, there are so many prediction techniques that are available in software engineering there is a need for stable software fault prediction methodology. In this research work, four types of Machine Learning techniques such as supervised, unsupervised, semi-supervised, and reinforcement learning are discussed. According to the study, to predict the fault, a fusion of classification and reduction Machine Learning technique is necessary. This report also introduces hypothesis set for fault prediction taxonomy.***

Keywords: ***Machine Learning (ML), Software Development Methodologies, Hypothesis, Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, Reinforcement Learning***

I. I INTRODUCTION

Nowadays software development are very multifaceted and has enormous dataset, so there is a necessity of early fault prediction. One third of the fault in software development to be identified early to minimize the rework. It is necessary to fix the likenesses between the faults of the newly developed software with that of the existing faults. There are lots of software development methods to develop a software. All the software methods begins with problem statement, requirements,collection, and development. After development, testing is used to verify and validate the developed software. Testing can be manual and also automated. This survey paper includes the analysis of various fault prediction methods. The aim of the paper is to derive the methodology to find the fault in the early stages so as to minimize the rework according to the customer view.

II. MACHINE LEARNING

Machine Learning techniques are the set of algorithms used by data scientists. The machine learning is to learn the mapping to make the prediction of Y for new X. It is represented as Y=f(X).The basic idea of machine learning is that parse data, learn from that data, and they apply what they have learned to make informed decisions. The basic structure of machine learning is described in the Diagram 1.

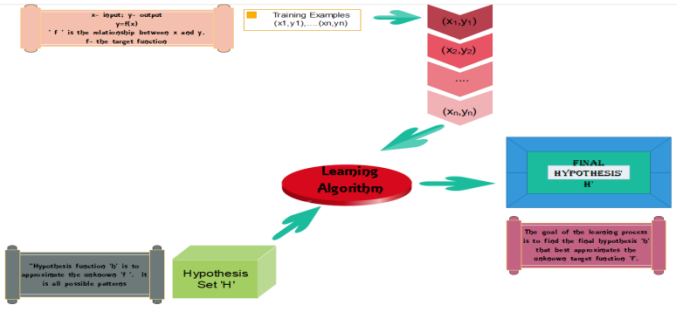


Diagram 1. Basic structure of Machine Learning

ML include five major steps for finding patterns for prediction. Those are data collection, data preparation, choosing appropriate algorithm and representation that is training a model, evaluating the model and performance improvement.

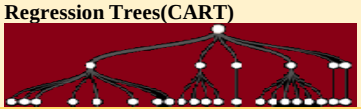
Algorithms used for Machine Learning

Machine learning is a subset of Artificial Intelligence.Machine learning means empowering the computer system with the ability to "learn". Basic ML models become progressively better but still need adjustments to make accurate prediction. Machine Learning has four types of learning methods. The methods are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised Learning has labels for training dataset. It helps to train the machines to learn the relationships between input and the result. In this the label is the known description given to the objects in the dataset. Unsupervised learning have no idea or knowledge about the resultant label. In this type, machines find patterns in the data by its own.

Revised Manuscript Received on March 25, 2019.
P. PATCHAIAMMAL, Research Scholar, Bharath University, Chennai
R. THIRUMALAISELVI, Assistant Professor, Computer Science Department, Govt. Arts College (Men), Nandanam, Chennai

Semi-supervised learning includes the combination of supervised and unsupervised learning types together. In this learning a part of the data is labeled and other part is raw unlabeled data. In this type, initial training set is loaded with labeled data. The model is trained on those data. Reinforcement learning includes a set of actions, parameters, and end values. It will train the machine by trial and error method. It will learn from past efforts to achieve the best possible result. The 10 popular Machine Learning algorithms are given in the below Table 1.

Table 1. Machine learning algorithms

ML category/Algorithm	Description	Technique used to learn
1. **Supervised learning-(Regression)** **Linear Regression** **Y=a*X + b** y-Dependent variable, a-slope X-Independent variable, b-Intercept	A line that best fits the relationship between the input (x) and the output (y) by finding specific weights.	Linear algebra solution for ordinary least squares and gradient descent optimization.
2. **Supervised learning-(feature extraction method)** **Linear Discriminant Analysis(LDA)**	Problem with more than two class classification. Predictions are made by calculating a discriminate value for each class.	Gaussian distribution
3. **Supervised learning-(Classification)** **Regression Trees(CART)** 	It is an implementation of decision trees. It is form of binary tree. Each node represents a single input variable (x) and a split point on that variable. The leaf node of the tree contain a result (y). It is used to make prediction.	Classification trees
4. **Supervised learning-(Classification)** **Naïve Bayes** **P(x|y)=(P(y|x)P(x))/P(y)**	It includes two probabilities one is probability of each class and the conditional probability for each class for given input (x).	Bayes theorem
5. **Supervised learning--** **K-Nearest Neighbors (KNN)**	For a new data entry the searching is done through entire training set for the K most similarities in the neighbors. Finally the output variables for K similarities are summarized.	Euclidean distance
6. **Supervised learning-(Classification)** **Support Vector Machines (SVM)**	Hyperplane is used for select the points in the input variable space by either 0 or 1 class. The points relevant in the hyperplane is added to the classifier.	Line (hyperplane)
7. **Supervised learning -(Classification and Regression)** **Random Forest**	Several samples are taken from the population to form models and the models are make predictions from that average is chosen for the better estimate of new data.	Bagging approach
8. **Unsupervised learning -(Clustering)** **K-Mean Clustering**	It works by finding patterns with grouped data. The groups are represented by 'K' variable.	Euclidean distance, centroids
9. **Supervised learning** **Neural network(NN)**	It has series of layers each of which are connected on either side.	Gradient descent
10. **Supervised learning(Classification)** **Logistic regression** $f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$ e-natural logarithm, x_0 - x- value of the sigmoid function ,L - curve's maximum value k-steepness of the curve	Problem with two class classification values used to find the values for the coefficients that weight each input variable.	Logistic function With values between 0 and 1.

Selecting the right algorithm is a key part of the Machine Learning. All the learning algorithms are used according to the problem and the data

III. LITERATURE SURVEY

Machine Learning uses techniques to find patterns in data. The patterns are then uses a model to recognizes it. Then the model used for making predictions in new data. The various ML techniques used in fault prediction is analyzed and listed in the Table 2.

Table 2 Literature Survey of Machine Learning in Fault Prediction

S. No.	Title and Author and Year	Algorithm/Methodology/Formula Used	Dataset used/Observed Result
	Nearest Neighbor Sampling for Better Defect Prediction – Gary D. Boetticher – 2005	Nearest Neighbor Approach 1. P (C = c_i | x) = P(C = c_i | A1 = a_{i1},…A_{in} = a_{in}) 2. PD = A / (A+B) 3. PF = C / (C + D) 4. Acc = (A + D) / (A + B + C + D)	NASA 1. Deeper understanding of defect data 2. Nearest neighbor test data distribution impact the test results

Retrieval Number: F1022376S19/19©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

	Enhance Rule Based Detection for Software Fault Prone Modules Hassan Najadat & Izzat Alsmadi January 2012	Data Mining – enhanced RIDOR (RIpple DOwn Rule) algorithm with CLIPPER algorithm $P(y,x) = \frac{\sum_{i:1\le i\le n\wedge F(xi)\ge}}{\mid\{i:1\le i\le}$ characteristic 2. Static code attributes with log filtration is used to improve accuracy and performance	NASA Public MDP (Modular toolkit for data processing)-Weka 1. Rule based classification method for fault prone
	Finding Defective Modules from Highly Unbalanced Datasets - J C Riquelme, R Ruiz, D Rodrıguez, and J Moreno - 2008	1. Random Over-Sampling (ROS) & Random Under-Sampling (RUS) 2. Synthetic Minority Oversampling Technique (SMOTE) 3. Decision tree classifier (C4.5) and a probabilistic classifier (Naïve Bayes)	PROMISE Dataset Balancing techniques may not improve the percentage of correctly classified instances, they do improve the AUC measure, i.e., they classify better those instances that belong to the minority class
	A survey on software fault detection based on different prediction approaches - Golnoush Abaei · Ali Selamat - November 2013	1. Method level Metrics 2. Two feature selection approach $\text{dist} = \sqrt{\sum_{i=1}^{n}(v_{1i}-v_{2i})^2}$ $\text{Affinity} = 1 - \text{dist}\,(\text{Ab}_k,\ \text{ep}_k)$	NASA dataset 1. Applying different feature selection techniques does not have that much effect on the results; they mainly reduce the execution time. 2. Different kinds of feature selection and method-level metrics do not have a considerable effect on the algorithm
	Cross Project Software Fault Prediction at Design Phase Pradeep Singh Shrish Verma (2015)	1. Naïve Bayes 2. True Positive Rate (TPR)=TP+(FN+TP) 3. False Positive Rate (FPR)=FP/(TN+FP) 4. ROC (Receiver Operating Characteristic) graph.	NASA dataset Area Under the ROC Curve (AUC) Metrics from the early software lifecycle are useful and should be used
	Genetic Feature Selection for Software Defect Prediction Romi Satria Wahono Nana Suryana Herman 2014	1. Genetic algorithm for feature selection 2. Bagging techniques for class imbalance problem 3. Fitness $fitness = W_A \times A + W_F \times \left(P + \left(\sum_{i=}^{n}\right.\right.$	NASA metric dataset AUC (Area Under Curve) used as the accuracy indicator for the performance of classifier. Proposed method makes an impressive improvement in prediction performance for most classifiers.
	Fault prediction using Early Life cycle Data	1. Probability of Detection(PD) $PD = \frac{TP}{TP+FN}$ 2. Probability of False Alarm(PF) $PF = \frac{FP}{FP+TN}$	NASA MDP dataset ROC curve used to represent the entire range of achievable performance characteristics relative to PD and PF. Statistical model was developed to predict defective software modules. Requirement metrics, module based code metrics, and the fusion of requirement and module metrics serve as predictors.
	Extracting software static defect models using data mining Ahmed H. Yousef September 2014	1. Naïve Bayes algorithm 2. Decision Tree algorithm 3. Precision = tp/(tp+fp) 4. Recall = tp/(tp+fn) 5. F-measure = 2 * Recall *recision/(Recall+Precision)	PROMISE Weighted voting rule approach is used to combine the decisions of the four models and achieved better results for accuracy, precision, recall and F-measure.

	Software Fault Prediction of Unlabeled Program Modules C. Catal U. Sevim B. Diri(2009)	1. Clustering 2. Metrics threshold 3. X-means Clustering method	Unsupervised software fault prediction can be fully automated and effective results can be produced by using X-means clustering with software metrics thresholds.
	Class level fault prediction Using Software Clustering Giuseppe Scanniello Carmine Gravino Andrian Marcus Tim Menzies(Nov 2013	1. Intra-release fault prediction technique 2. Software Clustering	PROMISE repository 1. Cluster rules do better than rules learned across the whole data. 2. In order to predict faults in a class is better to learn from classes related to it than from the entire system

From the above table we analyzed that there are various public datasets to identify the fault. Fault methodologies and tools are derived by using genetic algorithm and machine learning techniques. The metrics provide a best way for finding fault. Machine learning plays a vital role in finding the fault, but there is a need for the mixture of Supervised and Unsupervised Learning techniques such as classification and reductionto form the fault taxonomy in order to predict the fault more accurately.

V NEED FOR FAULT TAXONOMY

Fault is a flaw or condition that causes the software to fail to perform its function. Fault create failure of software system. Fixing fault in later life cycle of software development require rework. The rework require extra cost. The faultpercentage in software engineering stages shown in Diagram 2 describe the need for fault taxonomy. If one identify the fault at the later stage of software engineering the cost for identification and rework gets increased.

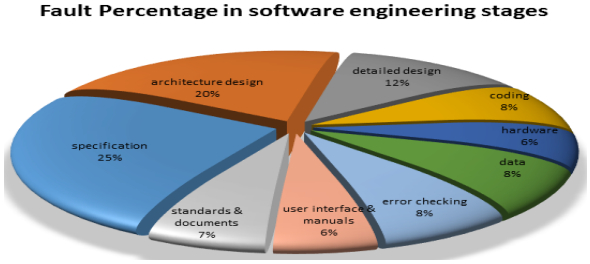


Diagram 2. FaultPercentage in software engineering stages

In the above diagram the software fault percentage of Project X from Company A is shown. According to the diagram the fault occurred percentage of software in each stages is like, specification 25%, architecture design 20%, detailed design 12%, coding 8%, data 8%, error checking 8%, standards and documents 7%, user interface and manuals has 6%, and hardware has 6%. It means finding fault in the final stage may increase the cost, rework and time.

The fault fixing cost spentinProjectX by Company A at each stage of software engineering is described in Diagram 3. The graph shows that fault fixing cost in the early stage of software engineering is less compared to later stages.

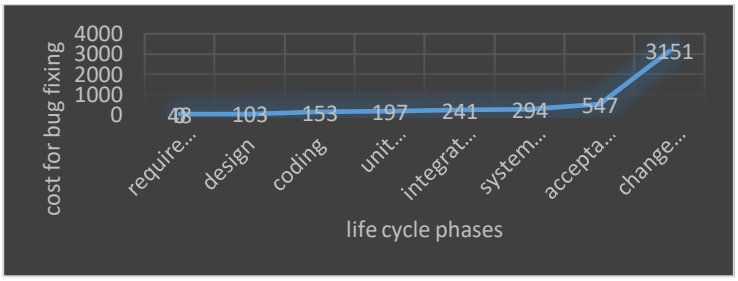


Diagram 3. Bug fixing cost in life cycle phases

From the above diagram we analyzed that fixing fault in the requirement stage is less compared after delivery stage. So finding the fault in the early stage is necessary to reduce the cost.

VI. HYPOTHESIS SET

Alearningalgorithm comes with ahypothesis space, the set of possible hypotheses. This hypotheses set is used to model the unknown target functions by formulating the functional setClassifier: A classifier is a special case of a hypothesis, which is nowadays often learned by a machine learningalgorithm.The hypothesis set formation for the fault prediction taxonomy by using already available labeled and unlabeled fault set. This hypothesis set is used to form fault taxonomy in for future developed software. By assuming previous fault dataset, this research survey is used to form the Hypothesis 'H1' to attempt to test the fault prediction in the software product.

H1: To predict the fault in a software, the combined classification and reduction ML techniques can be used.

VII. CONCLUSION

The development team needs to put extra efforts on finding the fault in the early stages to reduce rework in software production.

Retrieval Number: F1022376S19/19©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

There are lots of methodologies and techniquesthat are available to predict the fault. This paper analyzed various machine learning techniques used in fault prediction. The analysis report shows that the combination of machine learning techniques may produce better fault prediction methodology.It may also reduce cost and rework of software development at the same time. In this paper, Hypothesis set 'H1' is defined for the formation of fault prediction methodology. We conclude here that to get this considerable effect there is a need for better fusion machine learning prediction methodology.

REFERENCES

1. C. Catal, U. Sevim, B. Diri "Software Fault Prediction of Unlabeled ProgramModules" Proceedings of the World Congress on Engineering 2009 Vol IWCE 2009, July 1 - 3, 2009, London, U.K.
2. Felix Salfner, MarenLenk, and Miroslaw Malek,"A Survey of Online Failure Prediction Methods,"ACM Computing Surveys, Vol. 42, No. 3.
3. Yue Jiang, J. L., "Variance Analysis in SoftwareFault Prediction Models," IEEE 20th InternationalSymposium on Software Reliability Engineering.
4. Du Zhang., Jeffrey J.P. Tsai., "MachineLearning and Software Engineering". Software QualityJournal, 11, 87–119.
5. M. Hecht,K. S. Tso, S. Hochhauser "The Enhanced Condition Table Methodology for Verification of Fault Tolerant and other Critical Software"
6. Myron Hecht, Xuegao An, Bing Zhang, Yutao He "OFTT: A Fault Tolerance Middleware Toolkit for Process Monitoring and Control Windows NP applications"
7. Golnoush Abaei, Ali Selamat "A survey on software fault detection based on different predictionapproaches" - November 2013
8. Hassan Najadat, Izzat Alsmadi,"Enhance Rule Based Detection for Software Fault Prone Modules" January 2012
9. Yue Jiang, Bojan Cukic, Tim Menzies, Nick Bartlow "Comparing Design and Code Metrics for Software QualityPrediction" May 2008
10. T. Menzies, B. Caglayan, E. Kocaguneli, J. Krall, F. Peters, and B. Turhan, "The PROMISE Repository of empirical software engineering data," 2012. [Online]. Available: http://promisedata.googlecode.com
11. V. Basili, G. Caldiera, and D. H. Rombach, "The Goal Question Metric Paradigm, Encyclopedia of Software Engineering", John Wiley and Sons, 1994.
12. Ritika Sharma, N. B., "Study of Predicting Fault Prone Software Modules," International Journal of Advanced Research in Computer Science and Software Engineering.
13. Wasif Afzal, Richard Torkar, "On the application of Genetic Programming for Software Engineering Predictive Modeling: A Systematic Review," Expert Systems with Applications-An International Journal, March.
14. Cagatay Catal, Banu Diri, "A Fault DetectionStrategy for Software Projects" Technical Gazette, 20, 1,1-7.
15. Nurudeen Sherif, Nurudeen Mohammed "Using Fuzzy Clustering and Software Metrics to Predict Faults in large Industrial Software Systems" IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 13, Issue 6 (Jul. - Aug. 2013), PP 32-36
16. Rajesh, M., and J. M. Gnanasekar. "Path Observation Based Physical Routing Protocol for Wireless Ad Hoc Networks." Wireless Personal Communications 97.1 (2017): 1267-1289.
17. Rajesh, M., and J. M. Gnanasekar. "Sector Routing Protocol (SRP) in Ad-hoc Networks." Control Network and Complex Systems 5.7 (2015): 1-4.
18. Rajesh, M. "A Review on Excellence Analysis of Relationship Spur Advance in Wireless Ad Hoc Networks." International Journal of Pure and Applied Mathematics 118.9 (2018): 407-412.
19. Rajesh, M., et al. "SENSITIVE DATA SECURITY IN CLOUD COMPUTING AID OF DIFFERENT ENCRYPTION TECHNIQUES." Journal of Advanced Research in Dynamical and Control Systems 18.
20. Rajesh, M. "A signature based information security system for vitality proficient information accumulation in wireless sensor systems." International Journal of Pure and Applied Mathematics 118.9 (2018): 367-387.
21. Rajesh, M., K. Balasubramaniaswamy, and S. Aravindh. "MEBCK from Web using NLP Techniques." Computer Engineering and Intelligent Systems 6.8: 24-26.