International Journal of Innovative Science and Modern Engineering (IJISME)
ISSN: 2319-6386 (Online), Volume-6 Issue-6, April, 2020

Binary Class Classification of Software Faults in Software Modules using Popular Machine Learning Techniques

Devika S, Lekshmy P L

Abstract: ***Software engineering is an important area that deals with development and maintenance of software. After developing a software, it is always important to track its performance. One has to always see whether the software functions according to customer requirements. To ensure this, faulty and non- faulty modules must be identified. For this purpose, one can make use of a model for binary class classification of faults. Different technique's outputs differ in one or the other way with respect to the following: fault dataset used, complexity, classification algorithm implemented, etc. Various machine learning techniques can be used for this purpose. But this paper deals with the best classification algorithms available till date and they are decision tree, random forest, naive bayes and logistic regression (tree-based techniques and bayesian based techniques). The motive behind developing such a project is to identify the faulty modules within a software before the actual software testing takes place. As a result, the time consumed by testers or the workload of the testers can be reduced to an extent. This work is very well useful to those working in software industry and also to those people carrying out research in software engineering where the lifecycle of development of a software is discussed.***

Keywords: Software fault prediction, Decision Tree regression, software fault dataset, Machine Learning

I. INTRODUCTION

Software fault classification comes under the domain of software engineering. Software engineering deals with every part of development of software including making and testing of software. The actual classification of faults takes place before testing phase, thus helping the testers to have ease of work. While taking into consideration the concepts of machine learning, it is broadly classified as in Fig 1. Supervised and unsupervised learning is carried out in presence and absence of a supervisor respectively. Reinforcement learning on the other hand imitates human brain along with the concept of action and reward. Normally weka tool is used for this purpose. It includes inbuilt operations for different machine learning techniques. Out of the 4 techniques used, decision tree and random forest are tree-based learning techniques and naïve bayes and logistic regression are Bayesian-based learning techniques [8].

Revised Manuscript Received on April 22, 2019.
* Correspondence Author
Devika S*, Department of Computer Science and Engineering, LBS Institute of Technology for Women.
Lekshmy P L, Department of Computer Science and Engineering, LBS Institute of Technology for Women.

Also, fault prediction technique performs better when the percentage of software fault modules are less than a certain threshold value. In most cases, the threshold is fixed to be 30%. This work if executed properly will be helpful for researchers in the field of software fault prediction.

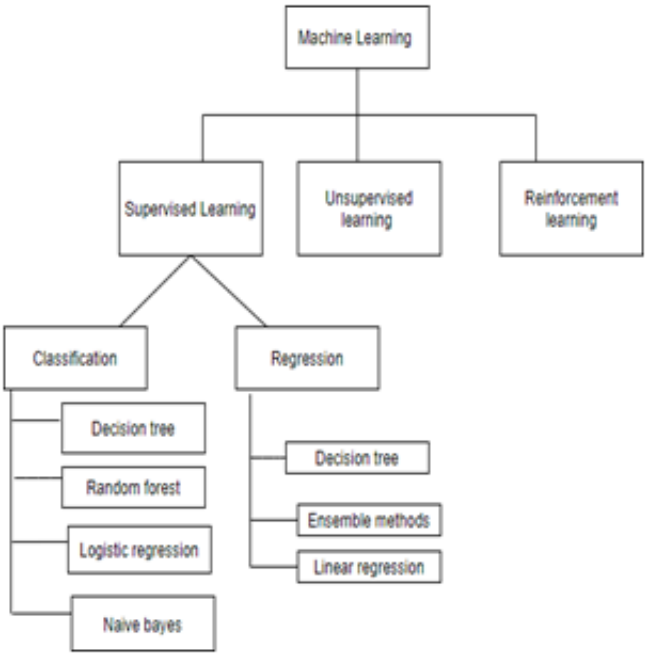

Fig. 1 Machine Learning classification

II. RELATED WORK

Devi et al. [4] proposed a feature selection model within the year 2012. The feature selection models used are Correlation based Feature Selection (CFS), OneR and Gain Ratio. Wrapper-based feature selection methods like naive bayes (NB), J48, etc. is additionally considered. OneR-NB gave the very best accuracy of 85.63%. It also helps to scale back complexity of classes thus providing good productivity and straightforward use of software. Rawat et al. [5] in the year 2012 prepared a literature survey on software quality improvement. The presence of software defects always degrades the performance of software. The famous data scientist Gaffney even deduced a formula that include lines of code to find out the number of defects. This work discusses the use of belief, genetic and neural networks. And belief networks overcame with most of the existing problems in defect management.

The use of Genetic Algorithms in a Web Fault Prediction with a live example is also discussed. Suresh et al. [2] published a work in the year 2014. This paper deals with the characteristics of dataset used. The dataset is collected from promise data repository. In this paper the author compares the performance of fault prediction of CK metrics dataset using predictive techniques such as logistic regression, linear regression, decision tree and Bayesian interface. Various CK metrics are - WMC, NOC (Number of Children), DIT, etc. are used. The performance evaluation parameters used are accuracy, completeness, correctness, etc. A graph is plotted against each of the performance measures to evaluate the final result. The coding is carried out in MATLAB. Out of all the CK metrics WMC appears to be the best for predicting fault. Also, a combination of techniques must be incorporated which is suggested as a future work. Mundanda et al. [1] released a work regarding software fault prediction in the year 2016. They discovered the effect of artificial neural network in software fault prediction. The model of ANN along with its predictive nature is also discussed. In this work, back propagation neural network is used. Gradient descent method is used for weight updating. The whole work is implemented using python. A 22 tuple promise dataset is used. It gave a preferable accuracy when compared to other models. But it resulted in different errors on datasets. Kaur et al. [3] published a survey work on software fault prediction the year 2018. The benefits and drawbacks of various machine learning techniques are discussed along with some of the performance measures. The difference among the terms error, fault, failure, etc are also discussed. The various forms of software fault control such as – software avoidance, tolerance and prediction techniques are also discussed. The discussed system is implemented using Fuzzy Interface System and it
metrics. And since we perform binary classification of faults, there must include a field mentioning faulty or non-faulty in our dataset. For our classification, promise data set Fig. 1 is used. The dataset can be downloaded from [10]. The only constrain to be considered while selecting dataset is that, the dataset must contain fault information. Here, the dataset includes information [6] about a software module being error prone or error-free. Table I [1] shows dataset attribute information for 22 tuples.

III. MACHINE LEARNING MODELS USED

Machine learning (ML) helps to learn, predict, decide, remember, analyze and recognize data. One of the major problems in ML is classification. Techniques used for classification problem are decision tree, random forest, naïve bayes, logistic regression.

A. DECISION TREE

Decision tree can be used for both classification as well as to determine a series of values. In classification, the tree will execute using if-then-else conditions e.g. the category to which a given fruit belongs. Regression tree is used when target variable is numerical [7]. For regression, a regression model is created and is given to all input or independent variables for prediction. One of the major advantages of decision tree is that its mechanism of tree split can be easily visualized by users thus making its operation easier. It can deal with both numerical as well as categorical data. It is never affected by non-linear parameters. The major disadvantages are overfitting, high variance in model, low bias. Here entropy is used to measure unpredictability in the dataset. As the tree is split into subgroups, information gain is found out, which is measure of decrease in entropy.

B. NAIVE BAYES

Naive bayes works on the property of bayes theorem for classification. Bayes theorem calculates the conditional probability of occurrence of an event using past history. Bayes theorem is as shown below.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where P(A|B) = probability of A provided B P(B|A) = probability of B provided A P(A) = probability of event A P(B) = probability of event B

also calculates performance measures like precision and recall.

Naïve bayes classifier is used to perform spam filtering, sentimental analysis in social media, text classification in online news channels. Advantages of using naïve bayes classifier are easy to implement, less training data required, deals both continuous and discrete values, best for real time predictors, not affected by irrelevant features.

IV. III. DATASETS USED

A software fault dataset is used for classification of software modules into buggy and buggy-free modules which helps to determine the faultiness present in modules of software. A software fault dataset includes both software metrics and fault information. The various characteristics of source code such as loc (line of code), v (volume), etc. are chosen as software

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

	loc	v(g)	ev(g)	iv(g)	n	v	l	d	i	e	b	t	lOCode	lOComment	lOBlank	locCodeAndComment	uniq_Op	uniq_Opnd	total_Op	total_Opnd	branchCount	defects
58	8	1	1	1	15	51.89	0.34	2.92	17.79	151.35	0.02	8.41	4	0	2	0	5	6	8	7	1	true
61	8	1	1	1	15	51.89	0.34	2.92	17.79	151.35	0.02	8.41	4	0	2	0	5	6	8	7	1	true
67	8	1	1	1	15	51.89	0.34	2.92	17.79	151.35	0.02	8.41	4	0	2	0	5	6	8	7	1	true
93	4	1	1	1	10	30	0.5	2	15	60	0.01	3.33	2	0	0	0	4	4	6	4	1	true
94	4	1	1	1	10	30	0.5	2	15	60	0.01	3.33	2	0	0	0	4	4	6	4	1	true
95	4	1	1	1	10	30	0.5	2	15	60	0.01	3.33	2	0	0	0	4	4	6	4	1	true
96	4	1	1	1	10	30	0.5	2	15	60	0.01	3.33	2	0	0	0	4	4	6	4	1	true
191	4	1	1	1	7	19.65	0.5	2	9.83	39.3	0.01	2.18	2	0	0	0	4	3	4	3	1	true
247	8	1	1	1	14	46.51	0.5	2	23.25	93.01	0.02	5.17	4	0	2	0	4	6	8	6	1	true
323	24	4	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	true

Fig 1. Promise data repository

Table I. Attribute information

Sl No.	Attribute	Explanation
1.	loc	line count
2.	v(g)	cyclomatic complexity
3.	ev(g)	essential complexity
4.	iv(g)	design complexity
5.	n	total operators + operands
6.	v	volume
7.	l	program length
8.	d	difficulty
9.	i	intelligence
10.	e	effort
11.	b	bugs
12.	t	time estimator
13.	lOCode	Line Count
14.	lOComment	count of comments
15.	lOBlank	count of blank lines
16.	lOCodeAndComment	Line Count + count of comments
17.	uniq_Op	unique operators present
19.	total_Op	total operators present
20.	total_Opnd	total operands present
21.	branchCount	related to flow graph
22.	defects	true, false

C. LOGISTIC REGRESSION

In logistic regression, one or more input variable is used to determine the binary output of the dependent variable. The response or output variable is always discrete in nature. It always results in a sigmoidal (S-shaped) graph.

$$\ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1$$

$$\Rightarrow P = \frac{e\beta 0 + \beta 1\, x}{1 + e\beta 0 + \beta 1\, x}$$

where P is the probability of happening of an event. β 0, β 1 are coefficients of line taken from equation of line, y = β 0 + β 1 x. Since we calculate the log of the probability, hence the name logistic regression. The exponential value is calculated so as to obtain the value of P.

RANDOM FOREST

Random forest is made by a collection of trees and the decision of majority of the trees is taken into consideration. Each branch of the tree represents a possible decision that is to be undertaken. Decision tree is a part of random forest. Main advantage of using random forest is its high accuracy, low training time during training phase and no overfitting. Random forest can even provide a considerable accuracy when the large amount of data is missing. Its major applications are in remote sensing, object detection, Kinect, etc. Random forest is better than all the above-mentioned techniques.

V. ARCHITECURE AND ALGORITHM

The architecture displayed in Fig 2. gives an overview of proposed approach. The input dataset used for the purpose is illustrated in detail in Section II. The machine learning techniques used for the classification are - decision tree, random forest, naive bayes, logistic regression. A detailed description of the techniques is discussed in Section III. The algorithm is given in Table II. It is coded in python [9].

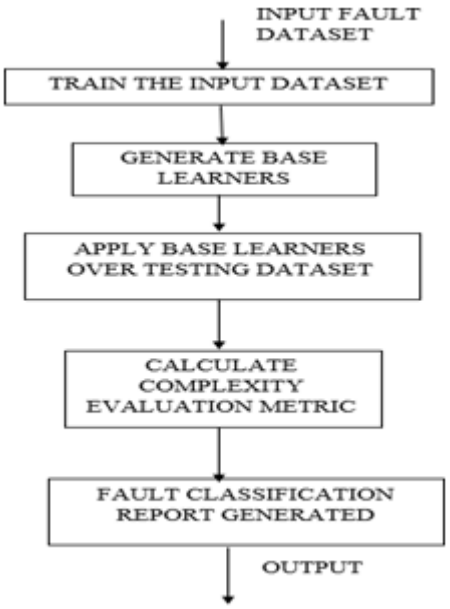

Fig 2. Overview of proposed approach

Table II. Algorithm

Step 1	Read the faulty dataset.
Step 2	Calculate complexity evaluation. All the values are benchmarks collected from software industries. Repeat through steps 2 (i) and (ii) for the entire datasets. i. Let A= if (total operators + operands<300) &(volume<1000) &(difficulty<10) &(effort<500000) & (time estimator<5000). ii. If (A = = True), complexity evaluation = successful, otherwise redesign.
Step 3	Split the given dataset into training and testing datasets.
Step 4	Let the input variables be the 1st 15 attributes and the output variable is the complexity evaluation metric.
Step 5	Now perform classification of faults using the four machine learning techniques

VI. EXPERIMENTAL SETUP AND RESULTS

Binary class classification of software faults can be carried out using the machine learning techniques mentioned in section I. At first, a heat map is drawn to find the correlation among various dependent variables in the data set. The figure given below, Fig. 3 shows the correlation. The flasher or light color in the heat map indicates that the covariance is high. The dark or dull color in the heat map indicates that the covariance is low. As a result, a high covariance is found between volume and bugs. After executing the algorithm mentioned in table II, maximum accuracy is received by decision tree, then random forest and then logistic regression. The least accuracy is received by naive bayes. Also, classification report of decision is given in table III. The final set of accuracies obtained by each technique is given in table IV.

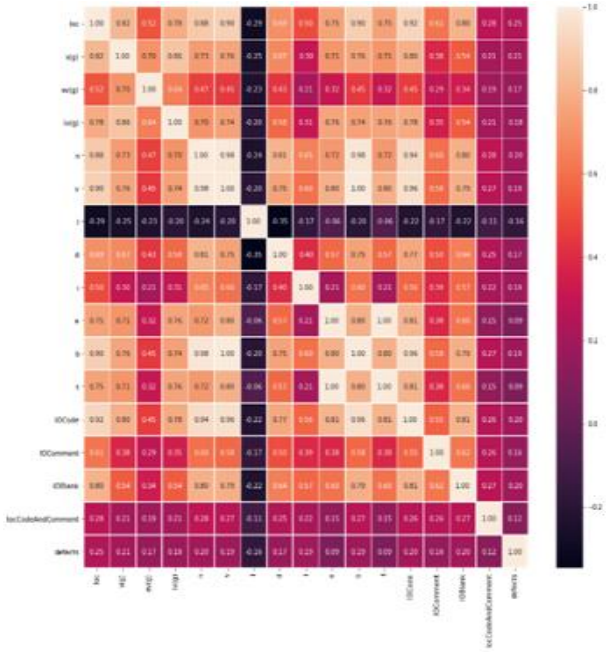

Fig. 3 Heat Map

Table III. Decision Tree Classification Report

R/S	Precision	recall	F1-score	support
Redesign	1.00	1.00	1.00	319
Successful	1.00	1.00	1.00	1858

Table IV. Final Result

Sl No.	Technique used	Accuracy
1.	Decision Tree (tree-based technique)	99.95%
2.	Random Forest (tree-based technique)	99.86%
3.	Logistic Regression (Bayesian technique)	98.80%
4.	Naive Bayes (Bayesian technique)	98.16%

VII. CONCLUSION

Software fault prediction popularly abbreviated as SFP, always tries to find out the fault- proneness in a software module. Different techniques have been used in this regard by varying input data, accuracy, etc. Based on the result analysis from previous studies, bayesian techniques like naive bayes, logistic regression and tree- based techniques like decision tree and random forest outperformed all other techniques and as a result these techniques where used for our work.

We were also able to conclude that, as the size of dataset decreases, Bayesian methods is said to produce better results and as the size of dataset increases, tree-based techniques gave better result. Apart from the carried-out technique, all other techniques differ with one another as a result comparison with one another is not possible. And it is to cure this problem that multiple base learners were used to classify faults. As a future work, prediction of fault count can be carried out instead of classification.

REFERENCES

1. Mundada, Devendra, et al. "Software fault prediction using artificial neural network and Resilient Back Propagation." *International Journal of Computer Science Engineering* 5.03 (2016).
2. Suresh, Yeresime, Lov Kumar, and Santanu Ku Rath. "Statistical and machine learning methods for software fault prediction using CK metric suite: a comparative analysis." *ISRN Software Engineering* 2014 (2014).
3. Kaur, Rajdeep, and Er Sumit Sharma. "Various techniques to detect and predict faults in software system: survey." *Int. J. Futur. Revolut. Comput. Sci. Commun. Eng.(IJFRSCE)* 4.2 (2018): 330-336.
4. Devi, C. Akalya, K. E. Kannammal, and B. Surendiran. "A hybrid feature selection model for software fault prediction." *Int. J. Comput. Sci. Appl* 2.2 (2012): 25-35.
5. Rawat, Mrinal Singh, and Sanjay Kumar Dubey. "Software defect prediction models for quality improvement: a literature study." *International Journal of Computer Science Issues (IJCSI)* 9.5 (2012): 288.
6. Kumar, Sandeep, and Santosh Singh Rathore. Software Fault Prediction: A Road Map. Springer Singapore, 2018.
7. Reena, P., and R. Binu. "Software Defect Prediction System–Decision Tree Algorithm With Two Level Data Pre-processing." *International Journal of Engineering Research & Technology (IJERT)* 3.3 (2014).
8. Kotsiantis, Sotiris B., Ioannis D. Zaharakis, and Panayiotis E. Pintelas. "Machine learning: a review of classification and combining techniques." *Artificial Intelligence Review* 26.3 (2006): 159-190.
9. Wójcicki, Bartlomiej, and Robert Dabrowski. "Applying Machine Learning to Software Fault Prediction." *e-Informatica Software Engineering Journal* 12.1 (2018).
10. https://datahub.io/machine-learning/jm1#data.

AUTHORS PROFILE

Devika S, is pursuing Master's Degree in Computer Science and Engineering from LBS Institute of Technology for Women, Kerala, India affiliated under Kerala Technical University. She received her Bachelor's degree in Computer Science and Engineering from Mohandas College of Engineering and Technology, Kerala, India in the year 2018, affiliated under Kerala University.

Lekshmy P L, received her Bachelor's degree in Information Technology from M S University, Tamil Nadu in 2004. Then she obtained her Master's degree in Computer Science and Engineering from Karunya Deemed University, Coimbatore in 2006. Currently, she is working as an Assistant Professor in Computer Science and Engineering, L B S Institute of Technology for Women, Trivandrum, University of Kerala (since 2008). Her current research interests are Privacy Preserving Datamining and Big Data analytics.