SOFTWARE FAULT DETECTION USING FUZZY C- MEANS AND SUPPORT VECTOR MACHINE

HIRALI AMRUTIYA[a1], RIDDHI KOTAK[b] AND MITTAL JOISER[c]

[abc]MEFGI, Rajkot, Gujarat, India

ABSTRACT

Organization like financial, medical, airline, and banking are require a very high quality software. If failure happen in this system cause high financial cost and affect the people lives. So it is important to develop the fault free software. Software fault detection is important for the software quality. Limited testing resources used to assurance the quality of software. The classification model is trained using the dataset. We tend to propose the framework which consist data pre-processing approach with Support vector Machine (SVM) classifier. In Data pre-processing relevance analysis perform using feature raking and redundant feature remove using the Fuzzy C- Means clustering techniques.

KEYWORDS: Data Mining, Fuzzy C-Means, Pre-processing, Software Defect Detection, Support Vector Machine

Software play important role in daily lives and become common day by day. Every one use the software in daily life so the quality of the software is most important for end user. Organization like financial, medical, airline, banking are require a very high quality software. If failure happen in this system cause high financial cost and affect the people lives. It's important to develop the quality software by organization or developer. Failure in software happen because of the ambiguities in requirements, design, code and testing cases. Fault identification in testing phase of software development life cycle not cost more but identification of failure at maintenance stage cost more. Software defect detection model is used for identifying the faulty component. This model used the limited testing resources for prediction of the fault.

Software metrics used for the detection of the fault. Metrics compute from the source code. Such metrics are size metrics, object-oriented metrics and complexity metrics. Different data mining techniques are used such as Neural Network, Genetic programming, Naïve Bayes approach, artificial immune system, fuzzy logics, decision trees. Supervised learner used with pre-processing or without pre- processing give different result. Pre-processing of the data increase the result of learning algorithm.

The benefits of the software defect detection model identify the fault before the testing so improve the software testing and software quality by allocating more resource on fault-prone module. Public NASA datasets are used for software defect detection .PROMISE repository, includes several public NASA datasets. [3]. Model used historical data for training the classifier.

The main objective of paper is to design software defect detection system using the fuzzy c-means and Support Vector Machine (SVM). Data pre-processing perform to identify relevant feature. Classification of fault data measure using the accuracy, ROC ,Area of ROC curve, recall, precision.

RELATED WORKS

The quality of dataset is improved by data pre-processing, which incorporates feature selection and sampling which reduce instances. In feature selection is the method of distinctive and removing irrelevant and duplicate features from a dataset in order that which increase the performance of classification model. [1]-[2]

Wangshu Liu et al. used data pre-processing approach which consists the two stage used for identify the fault in software. Compare result using the Naïve Bayes, C4.5, IB1.[3] Rohit Mahajan et al. proposed the framework for software fault prediction using the Bayesian Regularization (BR) and compare with Levenberg-Marquardt (LM) and Back propagation (BPA). Bayesian regularization give better performance. [4] Gupta, Deepika, Vivek K. Goyal, and Harish Mittal Proposed Estimating of Software Quality with Clustering Techniques. This paper focus on clustering with very large dataset and very many attribute of different types. Effective result can be produced by using fuzzy c-mean clustering. [5] Arashdeep Kaur et al. Proposed model for software fault prediction. In this paper, investigate the fuzzy c mean and k-mean performance. Fuzzy c mean is better than k- mean for requirement and combination metric model. Also investigate the metrics used in early life cycle can be used to predict fault module or not.[6] Wangshu Liu et al. use the clustering based feature selection method for software fault prediction. FF-relevance and FF- Correlation Measure use. Heuristics approach use for cluster formation. [7] Issam H. Laradji et al. use the greedy forward feature selection and Average Probability Ensemble learning model is to

[1]Corresponding author

classify data. This model contain seven algorithm such as W-SVM, Random Forest etc. [8]

PROPOSED FRAMEWORK

The fig 1 shows the proposed framework of system. The framework consist the two stage first stage is relevance analysis and second stage is redundancy control. Support vector Machine used as classification model.

Relevance Analysis

The relevance analysis stage identify the relevant feature from the dataset. For that feature ranking method is used. Correlation between the feature and class measure to find the relevancy. For measuring the correlation information gain (IG) use. Information gain measure the amount of information provided by the feature f, whether instance is fault or non-fault. The formula used for measuring the information gain is: [3]

$$IG(f) = H(A) - H(A|B) \quad (1)$$

Where H(A) compute the entropy of the district random variable A (i.e. class). Consider p (a) denote prior probability of a value a of A then H(A) compute by formula:

$$H(A) = -\sum_{a \in A} p(a)\log_2 p(a) \quad (2)$$

H (A|B) compute the conditional entropy which quantifies the uncertainty of A given the observed variable B. Consider p (a|b) denote posterior probability of a for value b. H (A|B) compute by formula:

$$H(A|B) = -\sum_{b \in B} p(b) \sum_{a \in A} p(a|b) \log_2 p(a|b) \quad (3)$$

Redundancy Control

Redundancy control stage is used for remove the redundant feature. For remove the redundant feature Fuzzy C- means algorithm is used. Fuzzy c-mean clustering moves the data centre iteratively to the right location.

Classification Model

Support vector machine (SVM) used as the classifier for the software defect detection. Support vector machine is supervised learning algorithm. In SVM plot data item in n- dimensional space with the value of each feature belong to the coordinate. Then classify the hyper plan to distinguish the class.

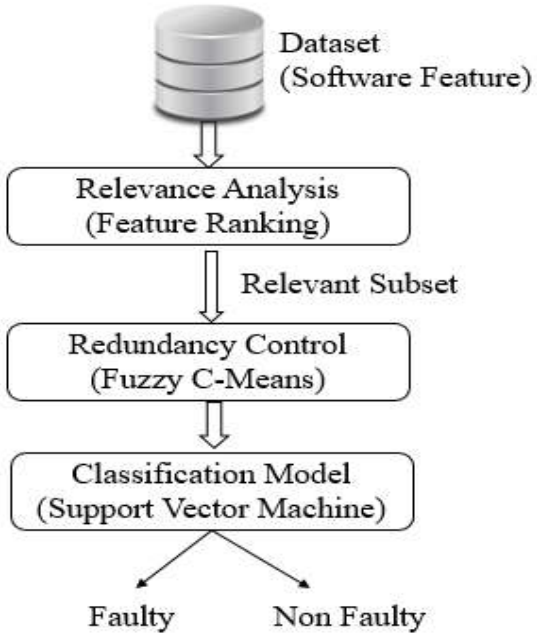

Figure 1: Framework of approach

DATASET AND TOOLS

Matlab 2010 used for the classification process. Waikato Environment for Knowledge Analysis (Weka) 3.17 used for the relevance analysis.

To evaluate the proposed framework we used the real world software NASA and Eclipse dataset. Dataset are freely available online. Software Dataset have software metrics as the feature. The different software metrics are:

Size Metrics

1. LOC(Line of code):Measure the productivity of the software related to length of the program. LOC count total length of the program including comment, statement, Blank Space. i.e. the higher the LOC higher the bug density.
2. Number of Statement: Measure the number of statement in the program. Statement include branching statement such as if, switch ,looping statement such as while, for, do-while, break and continue statement, try-catch and finally statement, method calls, return.
3. Number of Comment: Measure the number of comment which contain both single and multiple line comment. Program consists the 30% -75% comment then it is called good program. If comment less 30% means program is poorly explain.

Complexity Metrics

1. Halstead Complexity: Halstead metrics measure the operator and operands of source code. Halstead metrics depend upon the actual implementation of the program and its measure, which compute from the operator and operands of source code. The measures are:

N1=total number of operators N2=total number of operands n1=total number of distinct operators n2=total number of distinct operands

From this measure, several measure can be calculated, program length, volume, difficulty and effort.

2. McCabe Complexity: It is quantitative measure of linear independent path from source code. Control flow graph generated from the program code. For a program control graph G, Cyclomatic number (CC), is given as:

$$CC = E - N + P \quad (4)$$

Where, E = number of edges N = number of nodes

P = number of connected parts in graphs

Object Oriented Metrics

This metrics are used for object oriented language.

a. WMC (weighted method per class): Number of method per class is count in this metrics.

b. DIT (Depth of Inheritance Tree): depth of the class hierarchy is measure. Depth of the hierarchy is more than it is more complex to predict class behavior.

c. NOC (Number of children): This metrics measures number of direct subclass of the class.

d. CBO (Coupling between Object classes): Measures the number of other classes that class has coupled. Coupling between classes occur via return types, method call and inheritance.

e. RFC (Response for a Class): This metrics count number of method executed by the class object.

PERFORMANCE EVALUATION

The main aim of most classifiers is to perform binary classification, i.e., Faulty or Non-Faulty. The perform measure use are accuracy, confusion metrics, Area under the ROC curve.

Accuracy

Accuracy of the classifier means to correctly predict the class label of new or unseen data. Accuracy is percentages of testing set example correctly classified into class.

Area under ROC Curve (AUC)

Accuracy of the classifier means to correctly predict the class label of new or unseen data. Accuracy is percentages of testing set example correctly classified into class. For measure the area under the curve Receiver operating characteristics (ROC) is plot. Receiver operating characteristics (ROC) curve is graphical representation of the performance of binary classifier. The curve is created by true positive rate against the false positive rate at different threshold value. AUC is give better result for software defect detection.

Confusion Matrix

Confusion matrix used for measure the performance of classifier. Confusion table represent the true value identify from the set of test data. Confusion matrix has for basic categories which are True positive, True Negative, False Positive, False Negative. Table I represent the confusion table. From the confusion table another evaluation Measures Count which are:

1. Recall: It also called probability of detection or Sensitivity or true positive rate. Defined as the probability of correctly classified faulty module.

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \quad (5)$$

2. False Positive Ratio (FPR): It is also call fall-out. Defined as the ratio of false positive to non- fault module

$$\text{FPR} = \frac{\text{false positive}}{\text{false positive} + \text{true negative}} \quad (6)$$

3. Precision or positive predictive ratio (PPV): precision Measure the exactness of the classifier. It represent the percentage of the tuples that classifier labelled as faulty is actually faulty

$$\text{PPV} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \quad (7)$$

Table I. Confusion Matrix

		Actual Class	
Predicted		**Fault**	**Non fault**
	Fault	True positive	False positive
	Non fault	False negative	True negative

4. True Negative Rate (TNR): True negative rate measures the percentage of negatives that are correctly identify as such.it is also called specificity

$$\text{TNR} = \frac{\text{true negative}}{\text{true nigative} + \text{false positive}} \quad (8)$$

In this paper software defect detection system is implemented using Fuzzy C- Means clustering and Support vector machine. Pre- processing of data improve the Efficiency of the algorithm so feature ranking used ad pre- processing of data. NASA and eclipse public dataset are used for the system.

REFERENCES

Gao K., Khoshgoftaar T. M., Wang H. and Seliya N., 2011. "Choosing software metrics for defect prediction: An investigation on feature selection techniques," Softw.-Practice Exper., **41**(5):579–606.

Shivaji E. J. W. Jr., Akella R. and Kim S., 2013. "Reducing features to improve code change-based bug prediction," IEEE Trans. Softw. Eng., **39**(4):552–569.

Chen J., Liu S., Liu W., Chen X., Gu Q. and Chen D., 2016. "Empirical Studies of a Two-Stage Data Preprocessing Approach for Software Fault Prediction" in IEEE Trans. on Reliability, **65**:38-53.

Mahajan R., Gupta S.K. and Bedi R.K., 2014. "Design Of Software Fault Prediction Model Using BR Technique," in Proc. Int. Conf. Information and Communication Technologies, Kochi, pp. 849-858.

Gupta D., Goyal V.K. and Mittal H., 2013. "Estimating of Software Quality with Clustering Techniques," in Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on. IEEE.

Kaur A., Brar A.S. and Sandhu P.S., 2010. "An empirical approach for software fault prediction," in Industrial and Information Systems (ICIIS), International Conference on. IEEE, pp. 261-265.

Liu W., Chen X., Liu S., Chen D., Gu Q. and Chen J., 2014. "FECAR: A Feature Selection Framework for Software Defect Prediction," in proc. Int. Computers, Software & Applications Conference, pp. 426-435.

Laradji I.H., Alshayeb M. and Ghouti L., 2015. "Software defect prediction using ensemble learning on selected features", in Information and Software Technology, **58**:388-402.

Gupta D., Goyal V.K. and Mittal H., 2013. "Estimating of Software Quality with Clustering Techniques," in Advanced Computing and Communication Technologies (ACCT), Third International Conference on. IEEE, pp. 20-27.

Sharma R., Budhija N. and Singh B., 2012. "Study of predicting Fault Prone Software Modules," in International Journal of Advanced Research in Computer Science and Software Engineering, **2**(2):1-3.