An Efficient Technique for Software Fault Prediction in Variance Analysis

Anuradha S Deokar
M.Tech. Student
Department of Computer Engineering,
Bharati Vidyapeeth COE, Pune

V.M.Gaikwad
Asst. Professor
Department of Computer Engineering
Bharati Vidyapeeth COE,Pune

ABSTRACT

In this paper, we are using machine learning method for predicting fault, i.e support vector machine to predict the accuracy of the model predicted. The proposed models are validated using dataset collected from Open Source Software. The results are analyzed using Area under the Curve (AUC) obtained from Receiver Operating Characteristics (ROC) analysis. The results give you an idea about that the design predict by the support vector machine outperformed the entire the current models. Hence, based on these results it is reasonable to claim that quality models have a significant relevance with object oriented metrics and that machine learning methods have a Comparable performance with supervised learning methods.

Keywords

Support Vector Machine, Fault Prediction, Object Oriented, ROC curve

1. INTRODUCTION

Defect calculation model is basically double classifiers unfold by single of the supervised machine learning from sophistications technique as of moreover a separation of the defect. Information as of the present task otherwise the same as of a like history. Nowadays, machine learning is widely used in various domains (i.e., retail companies, financial institutions, bioinformatics, etc.) .There are various machine learning methods available and used to predict the accuracy of the model predicted. Variance be real assess of numerical Spreading of a random variable compute with averaging the squares of the deviations[7].When evaluating performance of a classification experiment, the smaller the variance, the more "reliable" (stable) the classifier performs[7]. Although the importance of variance in supervised classification is known, it is seldom reported and analyzed in software prediction models.

2. LITERATURE REVIEW

Significant work has been done in the field of fault detection. Recently, researchers have also started using some machine learning techniques to predict the model. Gyimothy et al. Calculated CK metrics from an open source web and email suite Fault Prediction by Statistical and Machine Learning Methods for Improving Software Quality call Mozilla. To validate the metrics, regression and machine learning method (decision tree and artificial neural networks) were used. The results whole NOC into the direction of be present not important but the entire the current metrics be initiate in the direction of be present strongly considerable Zhou et al. have used logistic regression and machine learning methods to show how object oriented metrics and fault proneness are related when fault severity is taken into account. The results were calculated using the CK metrics suite and were based on the public domain NASA dataset. WMC, CBO, and SLOC were found to be strong predictors across all severity levels. Prior to this study, no previous work had assessed severity of faults. During supervise twofold arrangement as well as information removal the dominate method designed for presentation study of a classifier be randomization along with cross-validation [9]. Randomization, although easy, be report by subsist the most reliable moreover unbiased system which have negative preconception. Breiman et al. demonstrate that k-fold cross validation have improved performance into requisites of falling variance. Kohavi show with the purpose of 10-fold CV can get improved bias-variance transaction by means of using resolution plants and NaiveBayes lying on a number of benchmark information set. Randomization and cross- validation be as fit the condition of expertise method in software work studies together with various example of experimental testing of defect calculation model [7]. On the other hand, deduction since miniature size defect calculation information set be able to direct towards inconvenience. Within prototype detection playing field Isakson et al. investigate the pitfall of cross-validation inside miniature test information categorization experiment .The recital key they used was error rate. Using Monte Carlo replication and a Parzen core thickness judgment classifier [1], Isaksson compare variances of inaccuracy speed designed in support of three changed trial size 20; 100, and 1000.They create to minor sample (20 and 100) contain big difference .The test of 1000 have a lot minor difference [7]. In direct towards reach consistent Categorization since a little numbers put, they suggest Repeated Independent Design and Test (RIDT) method, to be replicate experiment with via independent exercise and test separation various time and call offer experiments [10]. They furthermore propose

Towards details the final categorization outcome of little tests figures into the structure of Bayesian assurance gap which convey the variation evidently. In prior learn they used supervised binary classification used for recital testing of a classifier. In this study we are using machine learning methods to predict fault prone classes [5]. Results of various studies also show that better results are obtained with machine learning as compared to statistical methods.

3. SYSTEM DESCRIPTION

There are several stages while generating summary as shown in Figure 1.

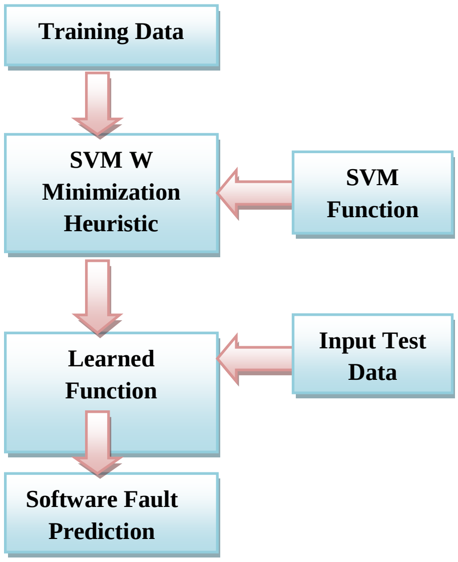

Fig 1.architecture diagram

Description of each of stage is given in the following section:

3.1 Training Data

In order to achieve this aim we have used dataset collected from open source software, poi. This software was developed using Java language and consists of 422 classes. The different dataset used by us will provide an important insight to researchers for identifying the significance of metrics through a given kind of dataset. As it is Open Source software, the users contain liberty to learn and adapt the basis system (write in Java) with no paying royalty towards preceding developers.

3.2 SVM Function

Basically Support Vector Machines are base on the idea of result plane to describe result borders. A result level surface is one that separate among a rest of items have dissimilar category memberships. A simple example is shown in the figure below. Here in this illustration, the items fit in each category GREEN or RED. The sorting out line defines a border line on top of the accurate region of which the entire items are GREEN in addition to the missing of which the entire items are RED. Some latest item (drawn round) decreasing towards the precise is labeled, i.e., classify, as GREEN (or classify as RED must it drops to the left of the sorting out stroke)

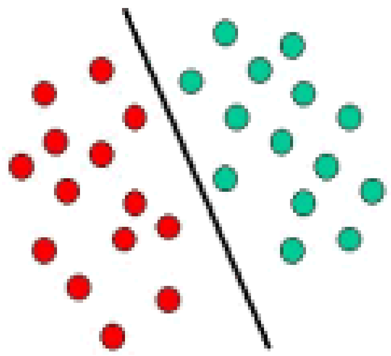

Fig 2. Line separate objects using red and green color

3.3 SVM W Minimization Heuristic

For SVM, instruction involves the minimization of the fault function:

$$\frac{1}{2}w^T w+C\sum_{i=1}^{N}\xi_i$$

focus to the constraints:

$$y_i\left(w^T\phi(x_i)+b\right)\geq 1-\xi_i \text{ and } \xi_i\geq 0, i=1,...,N$$

Any where C is the ability even w is the vector of coefficients, b is a stable and ξ_i represent parameter in support of use no separable information (input). The key i label the N preparation belongings .Make a message of to $y\in\pm 1$ represent the group label along with xi represent the free variables. The central part ϕ is used to convert data from the input (free) to the mark space. It must be distinguished that the outsized the C, the additional the fault exist penalize .Therefore, C be supposed to be selected through be concerned to pass up greater than correct.

3.4 Learned Function

The figure below show the fundamental thought follows Support Vector Machines. At this point we observe the unique items (gone part of the schematic) map, i.e., rearrange use a set of arithmetic functions, identified like kernel .The procedure of rearrange the items is recognized the same as map (alteration). Make a note of that so as towards into this new set, the map items (exact part of the schematic) is linearly separate and, therefore, as a substitute of construct the difficult curvature (absent schematic), the whole we include to carry out is to come across an best possible stroke to know how to divide the GREEN and the RED items. Classifying data is a familiar task in machine learning. Assume a quantity of set in arrange place each one is in the right place in the direction of single of two facilitate.

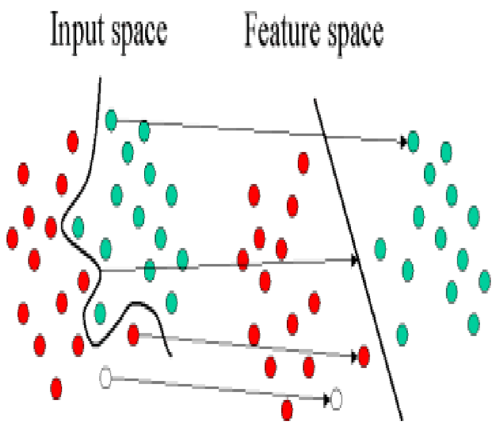

Fig 3. Linear SVM

The module along with the objective be headed for make a decision which group a latest facts head resolve subsist present here .During the casing of support vector machines, a facts indicate real analysis like p-dimensional vector (a listing of p facts), and we would like to distinguish whether we know how to separate such point among a (p − 1)-dimensional over-sensitive flat surface. It is called a linear classifier. At this point several lively planes so as to capacity categorize the information. Single sensible variety when the most excellent hyper plane is the distinct to represent the biggest partition, or else border among the two modules. So we want the hyper plane thus to facilitate the space beginning it towards the next numbers direct lying on every surface be maximize. But such a hyper plane exist it is well-known because the maximum-margin hyper plane. In addition to the linear classifier it define is recognized the same as a highest boundary classifier; otherwise consistently the view of most favorable strength.

4. METHODS AND MODELS USED

In this system following models are used and description each models are as below

4.1 Training of Software Fault Data

In this module a feature extraction algorithm is implemented which extracts the features from give dataset of software faults. These extracted features are use for classification of software.

4.2 SVM Mathematical Model

In this module support vector machine based mathematical model is used for classification.

4.3 Software Fault Prediction

In this module prediction of software fault is done using model built in module2 and feature extracted in module1.

5. AUC CALCULATOR

Receiver Operating Characteristics (ROC) curves are used to evaluate the performance of software fault prediction models [3]. This curve must pass through the points (0, 0) and (1, 1). The important regions of ROC curve are depicted in Figure 2. The ideal position on ROC curve is (0, 1) and no prediction error exists at this point. A line from (0, 0) to (1, 1) provides no information and therefore the area under ROC curve value (AUC) must be higher than 0.5[3]. If a negative curve occurs, this means that the performance of this classifier is not acceptable. A Software Fault Prediction Software fault prediction is one of the quality assurance activities in Software Quality Engineering such as formal verification, fault tolerance, inspection, and testing. Software metrics and fault data (faulty or non-faulty information) belonging to a previous software version are used to build the prediction model. The fault prediction process usually includes two consecutive steps: training and prediction. In the training phase, a prediction model is built with previous software metrics (class or method-level metrics) and fault data belonging to each software module. Following this phase, this model be use in the direction of calculate the fault-proneness label of module with the purpose of locate inside a new software version [5]. Recent advances in software fault prediction allow building defect predictors with a mean probability of detection of 71 percent and mean false alarm rates of 25 percent. These rates are at an acceptable level and this quality assurance activity is expected to quickly achieve widespread applicability in the software industry.

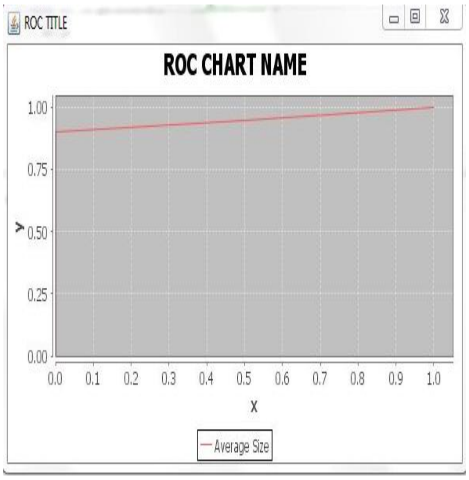

Fig.4 ROC Curve

6. CALCULATION OF CONFUSION METRICS

Model validation for machine learning algorithms should ensure that data were transformed to the model properly and the model represents the system with an acceptable accuracy [1]. There are several validation techniques for model validation and the best known one be N-fold cross-validation method [9] .This technique divide the dataset into N number of parts, and each of them consists of an equal number of samples from the original dataset. For each part, training is performed with (N-1) number of parts and the test is done with that part. Hall and Holmes suggested repeating these test M times to randomize the order each time. Order effect is a critical issue for performance evaluation because certain orderings can improve Degrade performance considerably. In table 1 confusion matrix is calculated after N*M cross-validation. Columns represent the prediction results and rows show the actual class labels. Faulty modules are represented label YES, and non-faulty modules are represented with the label NO. Therefore, diagonal elements (TN, TP) in Table 1 show the true predictions and the other elements (FN, FP) reflect the false predictions. For example, if a module is predicted as faulty (YES) even though it is a non-faulty (NO) module, this test result is added to the B cell in the table. Therefore, number B is incremented by 1. After M*N tests, A, B, C, and D values are calculated. In the next subsections, these values (A, B, C, D) will be used to compute the performance evaluation metrics.

Table 1. Confusion Matrix

	NO (Prediction)	YES (Prediction)
NO (Actual)	True Negative (TN) A	False Positive (FP) B
YES (Actual)	False Negative (FN) C	True Positive (TP) D

7. CONCLUSION

In this work we presented variance analysis on basis of support vector machine, performance is based on heuristics data. Variation is a significant reliability display of software fault prediction models. It reduces the software testing time. In the future, we can test more classifier and feature selection technique for fault prediction, to extend our work using different data set in biomedical field so that it can improve the performance of the system.

8. REFERENCES

[1] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7,2006.

[2] Y. Jiang, B. Cukic, and Y. Ma. Techniques for evaluating fault prediction models. *Empirical Software Engineering*, 13(5):561–595, 2008.[3] Y. Jiang, B. Cukic, and T. Menzies. Fault prediction using early lifecycle data. In *The 18th IEEE International Symposium on Software Reliability Engineering*, pages 237–246, Nov. 2007.

[3] S. Vanderlooy and E. Hullermeier. A critical analysis of variants of the auc. *Machine Learning*, 72:247–262, 2008.

[4] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[5] K. Wolter. *Introduction to Variance Estimation*. Springer Series in Statistics, 2007.T. M. Khoshgoftaar and E. B. Allen. Classification of fault prone software modules: Prior probabilities,costs, and model evaluation. *Empirical Software Engineering*, 3(3):275–298,1998.

[6] E. Arisholm, L. C. Briand, and E. B. Johannessen. A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. *Simula Technical Report*, TR-2008-06, 2008.

[7] Yue Jiang, Jie Lin, Bojan Cukic, Tim Menzies . Variance analysis in software fault prediction models.In The 20th International Symposium on Software Reliability Engineering, pages 99–108, Nov. 2009.

[8] Cagatay Catal, Performance Evaluation Metrics for Software, Acta Polytechnica Hungarica , Vol. 9, No. 4, 2012

[9] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36(1):111–148, 1974.

[10] A. Isaksson, M.Wallman, H. Goransson, and M. Gustafsson.Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recognition Letters*, 29:1960–1965, 2008.