International Journal for Rapid Research in Engineering Technology & Applied Science
ISSN (Online): 2455-4723
Volume II Issue X November-2016

DESIGN AND IMPLEMENTATION OF HYBRID APPROACH FOR IMPROVED SOFTWARE FAULT PREDICTION

Ankita Bajpai, Prof.Pritesh Jain
CSE Department, Patel College of Science and Technology, Indore, India*
CSE Department, Patel College of Science and Technology, Indore, India*
ankitabajpai1507@gmail.com, pritesh.arihant@gmail.com

Abstract:**The fault prediction approach contribution help throughout the software development by given that recourse to the present faults with the Bayesian nosiness.Machine learning classifiers have appeared as a method to predict the continuation of fault in the software. The classifier is primary trained on software narration data and then used to predict fault. The proposed system in excess ofcome the problem of possibledeficiency in accuracy for realistic use and use of a huge number of features. This paper suggests a feature selection technique appropriate to classification-based fault prediction. This approach is applied to predict faults in software codes and concert of Naive Bayes and Support Vector Machine (SVM) classifiers.**

Keywords: Bayesian Inference, Fault Prediction, SVM, Machine learning classifiers.

I. INTRODUCTION

Software testing is individual of the nearly allimportant, time consuming and dangerous quality declarationbehavior. It is a labor-intensive activity in software development life cycle while budget allocated for testing are usually limited [2], [3]. Software testing is a crucial activity during software development and fault prediction models support practitioners in this by as long asan upfront categorization of deficient software codes by illustration ahead the machine learning literature. While predominantly the Naive Bayes classifier is often practical in this observe, citing predictive performance and comprehensibility as its major strengths, a number of alternative Bayesian Algorithms that boost the possibility of constructing simpler networks with fewer nodes and arcs remain unexplored. This study contributes to the literature by considering different Bayesian Network (BN) classifiers and comparing them to other popular machine learning techniques. Furthermore, the applicability of the Markov blanket principle for feature selection, which is a natural extension to BN theory, is investigated. The results, both in terms of the AUC and the recently introduced H-measure, are rigorously tested using It is completed that easy and understandable networks with fewer nodes can be constructingwith BN classifiers other than the Naive Bayes classifier. In addition, it is establish that the aspect of directness and predictive performance require to be balanced out, and also the development context is an item which should be taken into during model selection.Work will be improved in these fields. First, to exam this method with more data. Besides only use one project's data is not convictive sufficient, dataset in dissimilar software project which center on dissimilar functions tends to closedissimilar weight of all matrix [3]. Secondly, to assess more resourcefulprocedure to discrete the dataset, naming data preprocessing, which is a important factor in present consequences. Finally, there have been several empirical studies that have examined the relationship of product metrics, failure history and fault proneness, but few that have explored the casual inference between them. The BN model provides a robust mechanism to detect the software defect prone.The paper is organized as follows: section 2 discusses the related work Section 3 proposed methodology 4 describes the result analysis based methodology.

II. RELATED WORK

In this paper, a comparative performance analysis of dissimilar machine learning techniques isinvestigate for software bug prediction on public obtainable data sets. Machine learning techniques are established to be constructive in terms of software faults prediction. The data from software repository include lots of information in evaluate software superiority and machine learning techniques can be useful on them in order to take out software faults information. The machine learning method is classified into two broad classes in order to evaluate their concert such as supervised learning versus unsupervised learning. In supervised learning algorithms such as all together classifier similar to bagging and boosting, Multilayer perception, Naive Bayes classifier, Support vector machine, Random Forest and Decision Trees are evaluate. In case of unsupervised learning technique like Radial base network function, cluster method such as K-means algorithm, K nearest neighbor are compared beside every one other.

III. PROPOSED METHODOLOGY

Software fault prediction is issue of main concern preventing a software system from errors is such a complicated task. There are forever quantities of changes that happen in Object oriented software system intend in continuous appearance. For cost reduction and improving the efficiency of software, it is extremely significant to classify the faulty module's software. In this study major focus is to get relative among software metric and faulty module. A software engineer should for eternity plan the changes for software design. Though it is extremely complicated to prefer the most excellent method for design but in our study new consequences illustrate that the proposed model can create the relation among software metrics and modules fault-proneness. A software fault prediction is an established technique in accomplish high software reliability. Software reliability can also be definite as the probability of failure-free software operation for a particular period of time in a particular environment. Prediction of fault-prone modules gives one way to support software quality. Quality of software is ever more important for software. For improving quality we must give more focus on testing for that portion of code which hasmajor number of faults. This study is an effort to predict fault in software by applydissimilar techniques. Though, this process necessitatesknowledge with some statistical model or machine learning technique [5].

PROPOSED ALGORITHM

The proposed hybrid algorithm generates a compilation of model (classifiers or predictors) for a learning scheme where every model gives an evenly weighted prediction.

Input:

D, a set of d training tuples;

K, the quantity of component in the collection;

A learning scheme (e.g., Naive Bayes Classifier, Support vector machine , etc.)

Output: A merged Component, M*.

Technique:

(1) For i = 1 to k do // generate k Component:

(2) generate bootstrap example, Di, by example D with substitute;

(3) utilize Di to obtain a Component,Mi;

(4) End for

To utilize the complex Component on a tuple, X:

(1) If categorization then

(2) Allow every of the k models classify X and return the mainstream vote;

(3) If prediction then

(4) Let every of the k Component predict a value for X and return the standard predicted value An significant advantage for combining superfluous and opposite classifiers is to increase sturdiness, accuracy and enhanced generally generalization. In this technique, the Support Vector Machine is construct technique is functional and assess error rate from the mean square error. Secondly, Support vector is performing with Support Vector Machine to get an extremely first-class overview performance.

Bayesian Network ClassifierNaïve Bayes Classifier: A Naive Bayes classifier is aeasy

probabilistic classifier based on apply Bayes'theorem (from Bayesian statistics) with strong (naive)sover eignty assumption. Aadd edexpressive term further essential probability replica would be independent feature model.In easy terms, a naive Bayes classifier assume that the being there (or absence) of exacting feature of a class is unconnected to the presence (or absence)of any additional feature. Even if these features depend on every other or ahead the continuation of the other features, a naive Bayes classifier consider every of these properties to separately contribute to the probability that this fruits ansoftware.Depending on the specific nature of the probability replica, naive Bayes classifiers can be trained tremendously proficiently in a Support vector machine. In lots of practical applications, parameter inference for naïve Bayes models use the technique of maximum likelihood inadditional words, one can effort with the naive Bayes model without believe in Bayesian probability or with any Bayesian methods .In spite of their naive design and in fact over-simplified assumption, naive Bayes classifiers have work moderately well in numerous complex real-world situations. [6]Analysis of the Bayesian classification problem has exposed that there are a number of theoretical reasons for the in factdifficultefficacy of naive Bayes classifiers. Still, acompleteevaluation with other classificationmethodsillustrate that Bayes classification is outperformed by further current technique, such as boost trees or random forests. An benefit of the naive Bayes classifier is that it merely require a little amount of training data to estimation the parameters (means and variances of the variables) essential for classification. Because independent variables are unspecified, only the variances of the variables for every class necessitate to be resolute and not the entire covariance matrix. The Naive Bayes Probabilistic Model conceptually, over a dependent class variable with a little number of outcome or classes, conditional on a number offeature variables.

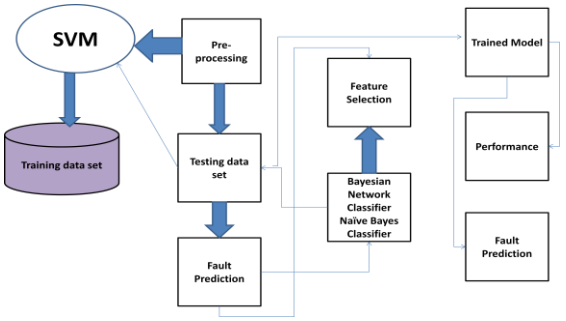

Figure 1: process of data processing in system

Support Vector Machines (SVM's) are a comparatively novel learning method used forbinary classification. The essential is to discover ahyperplane which divide the d-dimensional data completely into its two classes. However, since illustration data is often not linearly separable, SVM's initiate the concept of a kernel encourage feature gap which direct the data into a greater dimensional space where the data is distinguishable. Typically, casting into such a space would reason effort computationally and with in excess of appropriate. The key nearby used in SVM's is that the higher-dimensional space doesn't require to be dealt with straight (as it turns out, merely the method for the dot-product in that space is essential), which eradicate the above concern. In addition, the dimension of SVM's can be unequivocally calculated, unlike other learning methods like neural networks, for which there is no measure. Generally, SVM's are instinctive; theoretically well- founded and has exposed to be almost successful. SVM's have moreover been extended to resolve regression tasks (where the system is trained to yield a numerical value, relatively than yes/no classification) decision the optimal curve to the data is complicated and it would be a shame not to use the scheme of decision the optimal hyper plane. There is a method to pre-process.

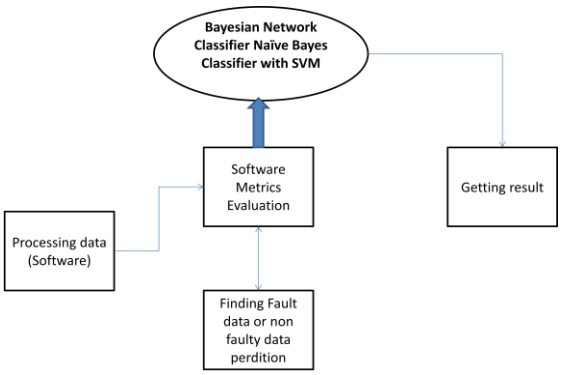

Figure 2: proposed data classification approach

The data in such a method that the problem is transformed into one of decision uncomplicated hyper plane. To do this, we describe a mapping z = (x) that transforms the d dimensional input vector x into a (frequently higher) d1 dimensional vector z. We expect to choose a () so that the novel training data.

Result analysis

In this section, to perform the experiment using tool just like dot net visual studio-2010 and sql server 2008.To used language C# and designing ASP.NET

the lag we assess our technique empirically with the metrics from the [1]data set. By analysing the error in software fault prediction, we study our representation in three dimensions, and compare it with exist work and Naïve-Bayes model.

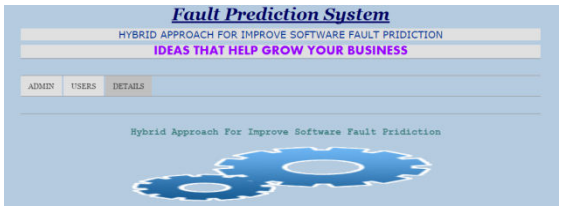

Figure 3: Fault prediction system

Communicate to table, the less arithmetical value of False Negative (FN) and True Negative (TN) stand for enhanced software quality. One of the objectives of this paper is to experimentally assess how Bayesian process can be used for assess software fault proneness. To demonstrate the evaluation between SVM's Naïve-Bayes model and our approach. The FN equals 0.0522 means extremely little experiential as non-failure modules are predict as a failure-prone module, while TN

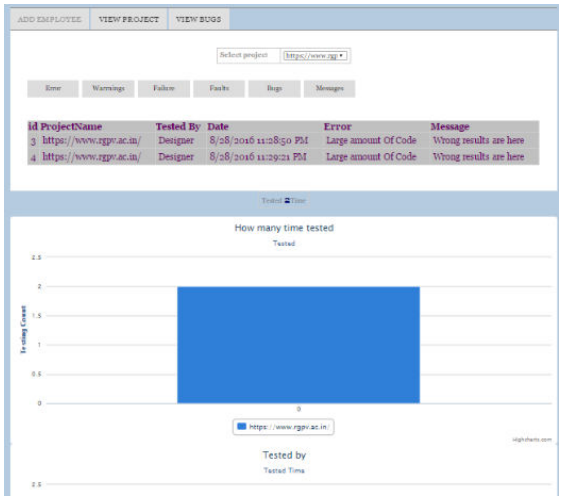

Figure 4: data extraction using our proposed system and check time complexity

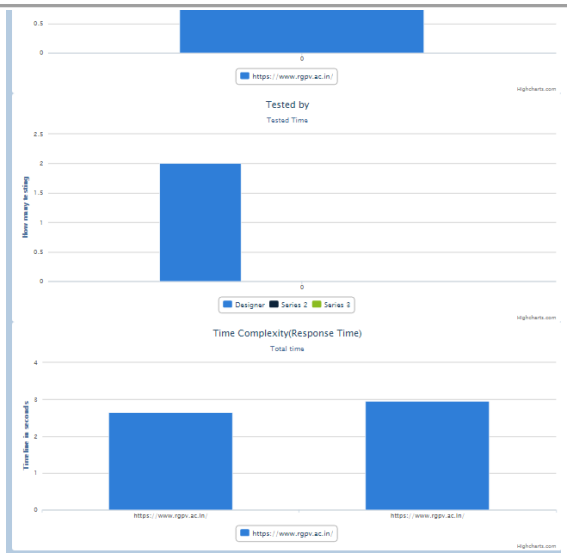

Figure 5: comparative graph exiting approach and proposed approach in term of time complexity response time.

FN rate in our approach is to a great extent less than their values, the Accuracy is advanced as well.

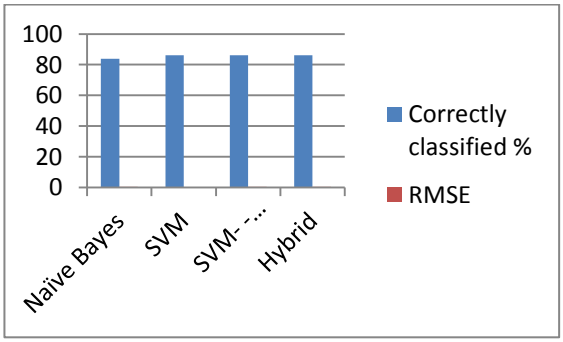

Figure 6: Classification Accuracy

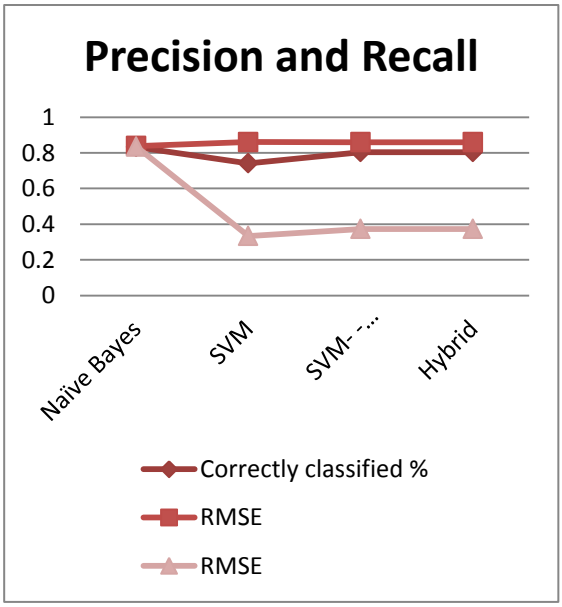

Figure 7: precision and recall

When evaluate with the Naïve-Bayes model, FN and Accuracy near précised.The software complexity process such as LOC compute, Cyclomatic complexity, dataset are use to classify the software component. Applying a machine learning technique written in Dot Net. It supports a number of data mining procedure such as pre-processing, clustering, classification and so on. All classification in this research is accepted Dot Net . For the performance estimate of the classifiers, Some samples from the taken live Dataset is used, wherein Some sample are used as training set and Some sample are use for testing.

IV. CONCLUSION

Our purpose in this research is to discoverexperientialconfirmation of the association among the bad smells and class error probability. From this study we intend Bayesian inference for entity metrics which give posterior probability for fault happening. Bayesian graph propose for the early onprediction of software fault is accessible in this paper. The model is based on software metrics and subsequent probability.Metrics have intended and with Bayesian inference system, predict probability of faults for subsequently piece of software. The Bayesian Inference replica is to classify posterior probability.Advance this learns can assist to identify threshold values of software metrics with receiver operating quality curves. We preparation to conduct added studies on open-source systems to discover out threshold values in recognize the faulty classes. For software developer, the model gives a methodology for assigning the resources for increasing reliable and cost-effective software.

Reference

[1] Raed Shatnawi1, Wei Li, James Swain," Finding software metrics threshold values using ROC curves",Journal Of Software Maintenance And Evolution: Research And Practice, Evol.: Res. Pract. 2010; 22:1–16

[2] Ganesh J. Pai, Member, IEEE, and Joanne BechtaDugan ," Empirical Analysis of Software Fault Content and Fault Proneness Using Bayesian Methods",ieee transactions on software engineering, vol. 33, no. 10, october 2007.

[3] Satwinder Singh and K.S. Kahlon ," Effectiveness of Encapsulation and Object-oriented Metrics to Refactor Code and Identify Error Prone Classes using Bad Smells", AcmSigsoftSoftware Engineering Notes Volume 36, Number 5,September 2011.

[4] HeenaKapila ,Satwindersingh "Analysis of CK Metrics to predict Software Fault-Proneness using Bayesian Inference" International Journal of Computer Applications (0975 – 8887) Volume 74– No.2, July 2013.

[5] Tracy Halla, Sarah Beechamb, David Bowesc, David Grayc, Steve Counsella, " A Systematic Review of Fault Prediction Performance in Software Engineering". IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO.6, 2012.

[6] K. Dejaeger, T. Verbraken, and B. Baesens, "Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 39, NO. 2, FEBRUARY 2013.

[7] RuchikaMalhotra and Ankita Jain, "Fault Prediction Using Statistical and Machine Learning Methods for Improving Software Quality", Journal of Information Processing Systems, Vol.8, No.2, June 2012.