

## A Technique for Improving Software Quality using Support Vector Machine

J. Devi<sup>1\*</sup>, N. Sehgal<sup>2</sup>

<sup>1</sup>Department of Computer Science engineering, Baddi University, Baddi, India

<sup>2</sup>Department of Computer Science engineering, Baddi University, Baddi, India

\*Corresponding Author: [devijyoti153@gmail.com](mailto:devijyoti153@gmail.com) Tel.: +91-8894812911

Available online at: [www.ijcseonline.org](http://www.ijcseonline.org)

Received: 27/May/2017, Revised: 06/Jun/2017, Accepted: 22/Jun/2017, Published: 30/Jun/2017

**Abstract**— Today software has reformed the key element on every environment. Quality of software is connected with the number of faults as well as it determinate by time and cost. Software is a process and maintains continuous change to improve the functionality and effectiveness of the software quality. During the life cycle of software various problems arises like advanced planning, well documentation and proper process control. Software defects are expensive in specification of cost and quality. Software defect prediction improves quality framework predictive techniques and software metrics to provide fault-prone module description. This paper main feature is the concept of change proneness and software prediction model used to control the classes of software which are often to change. We have two aspects to be inscribed Parameters like Accuracy, Precision, Recall and Receiver operating characteristics (ROC). Machine learning algorithms are used for predicting software. This paper is proposing to relate and compare all machine learning techniques interrelated to performance parameters.

**Keywords**—Software Quality, Support Vector Machine, Software Defect Prediction, Faults Prone, Change Proneness.

### I. INTRODUCTION

Software is a process and maintains continuous change to improve the functionality and effectiveness of the software quality. During the life cycle of software various problems arises like advanced planning, well documentation and proper process control. The software attributes are categorized two type's internal and external quality. The internal quality like efficiency, maintainability, testability, flexibility, reusability etc. The external quality like integrity, usability, reliability and accuracy etc [1,2,3,4,5,6,7,8,9, 10,11,12,13,14,15,16,17,18,19]. In this paper improving software quality by using machine learning classifier like a decision tree, naïve bayes and support vector machine. In present time machine learning used in the various fields like computers, military, medical, biomedical, engineering fields. In last few years machine learning fields is very popular. Machine learning "The ability of machine to improve the quality and performance based on previous results and data". There are various types of the machine learning like a supervised learning and unsupervised learning. The supervised learning a mapping from a set of inputs to a target variable.

Classification: target variable is discrete.

Regression: Target variable is real value.

The unsupervised no target variable provided. There are various types of classifier used in machine learning like naïve bayes, decision tree, Ridor, simple logistics, ordinal class classifier, voted perception, support vector machine and modified support vector machine.

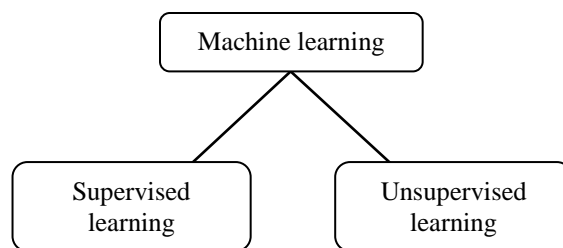


Figure. 1 machine learning

### II. RELATED WORK

In this section, we will focus on the research background which is required for the accomplishment of the proposed methodology. Firstly we will explain the use of machine learning classifiers.

Naïve bayes classifier has persisted commonly used for classification due to its simplicity and efficiency. It is a model based classification method and tender competitive classification performance for text categorization compared with other data driven classification method like SVM. The classification decision is built maximum a posteriori (MAP) rule. Usually three distribution model, including Bernoulli model, multinomial model and passion model, have commonly been incorporated in the Bayesian framework and have resulted in classifiers of Bernoulli naïve bayes, multinomial naïve bayes, and Poisson naïve respectively. For this reason, the naïve bayes sometimes refers to the MNB

classifiers. In this paper we focus on the generation of the proposed feature selection method now the MNB classifier. The methods can easily continue to BNB and PNB classifier [8,20,23].

Decision tree induction is one of the simplest forms of supervised learning classifier. It has been extensively used in many areas such as statistics and machine learning for the purposes of the classification and prediction. Decision tree classifiers can be generalize beyond the training sample so that unseen samples could be classified with the high accuracy as possible. Decision tree are non-parametric and a useful means of representing the logic embodied in software routines. A decision tree takes an input a case or example described by a set of attributes values, and outputs a Boolean decision. In the classification case, when the response variable takes value in a set of previously defined classes the node is assigned to the class which represents the highest proportion of observations [19, 21,24,25].

Support vector machine (SVM) in ordered to classify data points toward linear separable data set in 1998. It was first proposed by statistical theory for elucidate binary classification problems. Minimizing and upper bound of the errors and maximizing the distance between the separated hyper plane and data it tries to find the optimal hyper plane. It builds a maximal separating hyper plane to map input vector to a higher dimensional space. Two parallel hyper planes are built and the data are separated on each side of the hyper plane [11,26,27,28,29,30,31].

**III.PERFORMANCE METRICS MEASURES**

The performance of the algorithms can be measured conforming to assured metrics like accuracy, specificity, sensitivity etc. a confusion matrix form the basic from which different parameters can be calculated. The confusion matrix is always equated by four values which are TP, TN, FN, and FP as shown in fig 3. The parameters discussed in below.

True positive (TP): This case was positive predicted positive.

True negative (TN): This case was negative predicted negative.

False positive (FP): This case was negative but predicted positive.

False negative (FN): This case was positive but predicted negative.

ACTUAL CLASS	PREDICTED CLASS	
	Yes	No
Yes	A	B
No	C	D

Figure. 2 confusion matrix

A: TP (true positive)                      C: FP (false positive)  
 D: TN (true negative)                     B: FN (false negative)

Accuracy, Precision, Recall and ROC (receiver operating characteristics). The accuracy metrics is widely used in machine learning fields, which indicates the overall performance. The precision is check all positive predictions are correct. Precision is a measure of how many positive predictions were actual positive observations.

Precision= TP/ (TP+FP)

The recall checks all positive observation that is correct. The recall same as true positive rate (TPR).

Recall= TP/ (TP+FN)

Accuracy: Accuracy checks all the prediction is correct.

Accuracy =TP+TN/ (TP+TN+FP+FN)

ROC (receiver operating characteristics): In scale to design the curve separating true positive rate (TPR) and false positive rate (FPR) this caption is appropriate. The resulting curve represents the trade off between true positive rate and false positive rate. The area under curve the curve is termed as AUC that gives the value of ROC. The greater area the curve fill, largest will be the value of ROC. The area under the range from 0 to 1 and a property with more predictive power result in an area under the ROC closer to 1[23, 30]. A rough guide for classifying the accuracy of a system is the traditional academic point system:

- .90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

Specificity: It is also known as true negative rate and gives the measure of the real negative that is described precisely.

Specificity: TN/ (FP+TN)

Sensitivity: It is also known as true positive rate and gives the implication of the real positive that is correctly described.

Sensitivity: TP/ (TP+FN)

#### IV. METHODOLOGY

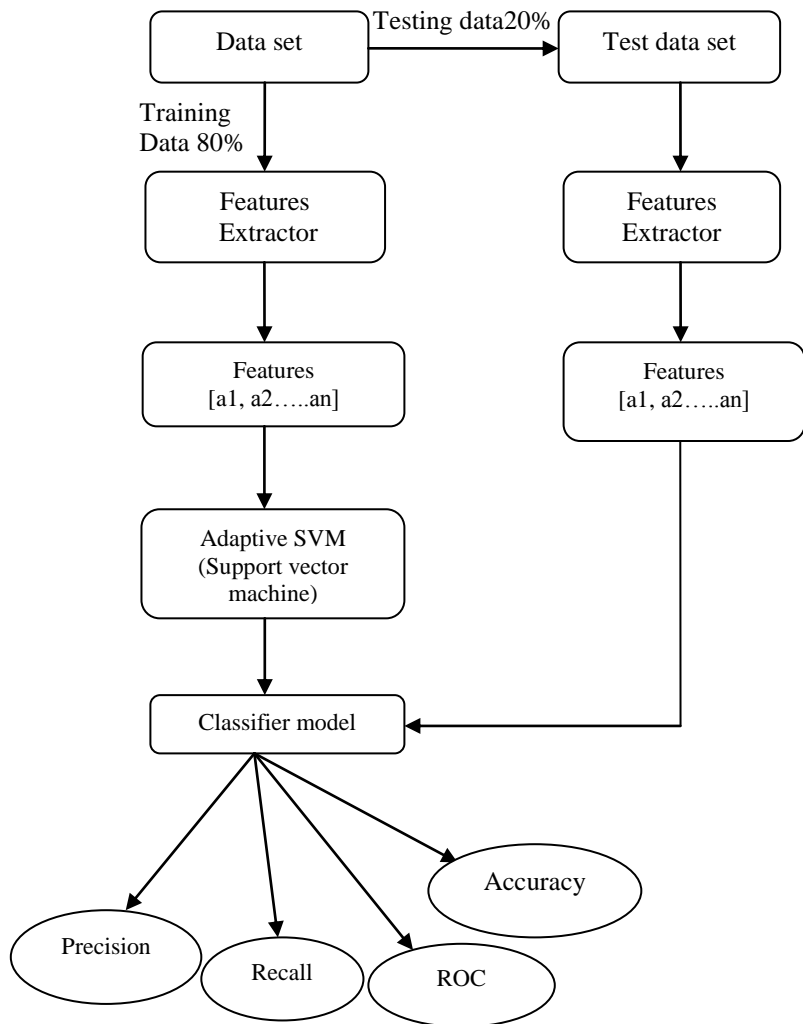


Figure. 3 Proposed methodologies

Step1: Android data set with different features like time estimator, fit model, effort and design complexity etc whereas defect prediction in software module.

Step 2: Implement features extractor on data set. Features extractor is used to merge the data set.

Step 3: Take on the different features and find out the status which if they are default or not. Default (a1, a2, ..., an) whenever the value is +1 which means its 'default' and if -1 then it is 'not default'.

Step 4: Implement machine learning classifier adaptive SVM.

Step 5: Apply classifier model to find out classifier precision, recall, accuracy and receiver operating characteristics (ROC) of the software module.

#### V. DATASET DESCRIPTION

Here, we consider and outcome dataset by the benefits of the data collection approach. We get checkout the theory of open source software that is specified written in the language java designate of as "android". The phrasing of the open source software acts basically the computer software with code and grant for this concede study promote and change it for part of useful destination.

Android software version dataset: 4.3.1, 4.4.2

Java tool dispense us a result analysis of 700 classes for overall with explication of any change in identification the two versions in circumstances of "yes" 0 or "no" 1. For the purpose of predicting model we get mat lab software. Mat lab is a suite of machine learning software written in java.

#### VI. EXPERIMENTAL RESULTS

In this relationship established over the analysis enclosed by change proneness of software classes and its metrics have being define. Along be apply part of the machine learning approach's to conduct the combine result of the software metrics of classes and the change proneness. The data points that are collected from the versions of the software that are used for standard predictions. Part of the measures that are used for appraise the performance of each predicted change proneness standard are explained.

Precision help we can check the all positive prediction that are correct. In this case checks the positive predictions are match the real result. Recall help we can check the real positive observation that are correct. Accuracy means that all the positive and negative prediction that is correct.

For predicting accuracy, we have applied the model to different sets of data like android 4.3.1 and 4.4.2. Firstly data set split into two parts training data set and testing data set. The training data set are divided 80% data set and test data are divided 20% data set. We can calculate the performance of software with the help of performance measures like precision, recall and accuracy.

#### VII. IMPLEMENTATION

In this paper, the use multiple measure like precision, recall, accuracy, roc area. In this measures help we check the performance in terms of accuracy and quality. In machine learning, checks the performance in terms of number of positive values are correct, number of predicted values are

correct and number negative values are correctly predicted. In previous work, Naïve bayes and decision tree already used but performance is low in terms of accuracy. In proposed work support vector machine (svm) and modified svm used to improve the performance in terms of accuracy.

The result achieve from mat lab is shown in the table.1:

Name of algorithm	Precision	Recall	Accuracy	Roc area
Naïve bayes	0.97	0.9923	0.9809	0.9925
Decision tree	0.969	0.9923	0.9761	0.9879
Support vector machine	0.992	0.9923	0.9904	0.9996
Modified support vector machine	1	1	1	1

Table. 1

Fig. (4) Represent the value of naive bayes in terms of accuracy, quality.

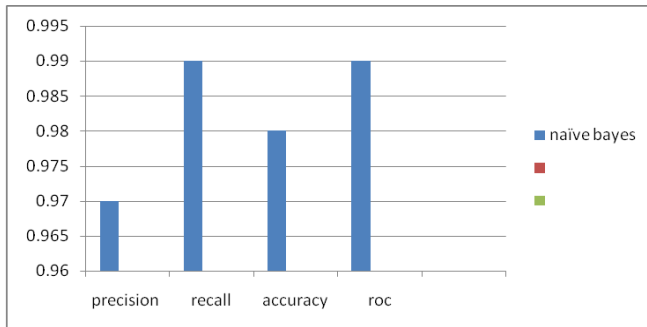


Figure. 4 graphical representations of naive bayes

Fig. (5) Represent the value of decision tree in terms of accuracy and quality.

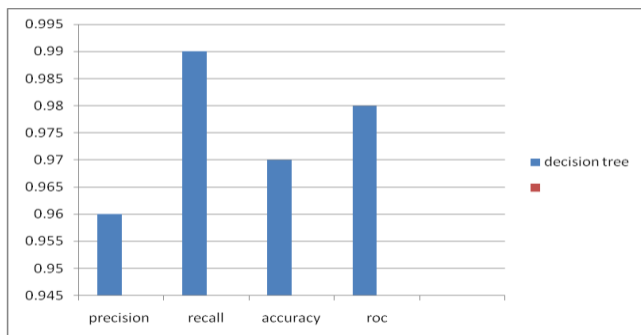


Figure. 5 graphical representation of decision tree

Fig. (6) Represent the value svm in terms of accuracy and quality.

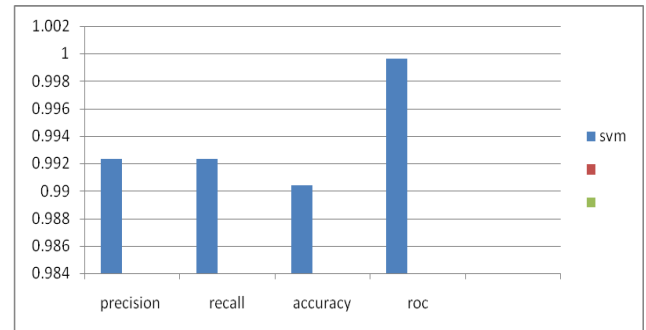


Figure. 6 graphical representation of svm

Fig. (7) Represent the value modified svm in terms of accuracy and quality.

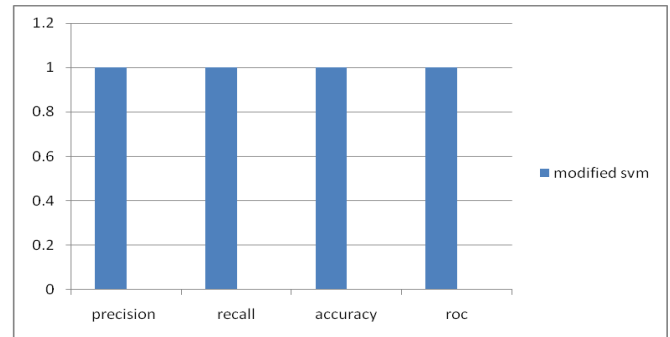


Figure.7. graphical representation of modified svm

Comparison of different classifier:

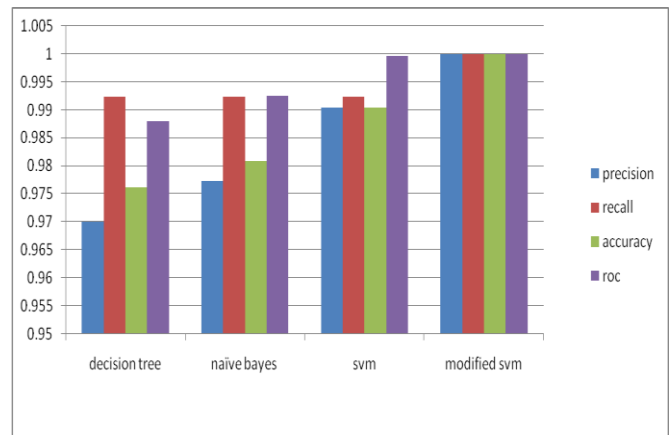


Figure 8. 8Graphical implementation of different classifier results

### VIII. CONCLUSION

Prediction provides predicting precise/continuous value for input. However, machine learning techniques like decision

tree, naïve bayes and support vector machine (SVM) are modifiable for prediction. These algorithms help to improve the performance of the software quality and performance. In terms of the accuracy, precision, recall and receiver operating system (ROC). However, the existing work over fitting and data scarcity issue. To overcome this problem an algorithm is proposed using modified SVM to enhance the performance and get better result in term of accuracy and performance. The software performance check with the help of performance measures. Our goal is to reduce the testing time as with these techniques.

### IX. FUTURE SCOPE

In our work, we have related and compare all the machine learning techniques. We have also compared other android version datasets with the help of different classifiers. Advance different algorithms can implement and invented. Lesser time waste in the training and testing. We have also work in improving training and testing is acute and effective way to manage.

### REFERENCES

- [1]. K. Chandra, "Improving software quality using machine learning", Innovation and Challenges in Cyber Security, India, pp.115-118, 2016.
- [2]. B. Tang, "Toward optimal feature selection in naïve Bayes for text categorization", IEEE Transactions on Knowledge and Data Engineering, vol.5, issue.9, pp.2508-2521, 2016.
- [3]. S. Choudhury, "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection", Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials IEEE, India, pp.89-95, 2015.
- [4]. L. Zhao, "Quality Evaluation of College Foreign Language Textbook Based on SVM", Intelligent Computation Technology and Automation IEEE, China, pp.670-673, 2015.
- [5]. L. Zhao, "Quality Evaluation of College Foreign Language Textbook Based on SVM", Intelligent Computation Technology and Automation IEEE, China, pp.670-673, 2015.
- [6]. C. Izurieta, "Preemptive management of model driven technical debt for improving software quality", Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures, USA, pp.31-36, 2015.
- [7]. A.D Lipitakis, "On machine learning with imbalanced data and research quality evaluation methodologies", Computational Science and Computational Intelligence International Conference IEEE, USA, pp.451-457, 2014.
- [8]. G. Vitello, "A Novel Technique for Fingerprint Classification based on Fuzzy C-Means and Naive Bayes Classifier", Complex, Intelligent and Software Intensive Systems IEEE, UK, pp.155-166, 2014.
- [9]. T.M Khoshgoftaar, "Improving software quality estimation by combining feature selection strategies with sampled ensemble learning", Information Reuse and Integration 15th International Conference on IEEE, USA, pp.428-433, 2014.
- [10]. Y. Yibo, "Scalable svm-based classification in dynamic graphs", Data Mining IEEE International Conference on. IEEE, China, pp.650-659, 2014.
- [11]. Y. Chen, "Effective Part Localization in Latent-SVM Training", Pattern Recognition 22nd International Conference on. IEEE, Sweden, pp.4269-4274, 2014.
- [12]. S. Mahajan, "Hierarchical Reinforcement Learning in Complex Learning Problems: A Survey", International Journal of Computer Sciences and Engineering, Vol.2, Issue.5, pp.72-78, 2014.
- [13]. E. Shihab, "Practical software quality prediction" Software Maintenance and Evolution IEEE International Conference on. IEEE, Canada, pp.639-644, 2014.
- [14]. M. Shepperd, "Researcher bias: The use of machine learning in software defect prediction", IEEE Transactions on Software Engineering, vol.40, issue.6, pp.603-613, 2014.
- [15]. P. A Selvaraj, "Support Vector Machine for Software Defect Prediction", International Journal of Engineering & Technology Research, pp.68-76, 2013.
- [16]. D. Gray, "The misuse of the NASA metrics data program data sets for automated software defect prediction", Evaluation & Assessment in Software Engineering 15th Annual Conference on IET, UK, pp.603-613, 2011.
- [17]. Q. Song, "A general software defect-proneness prediction framework" IEEE Transactions on Software Engineering, vol.37, issue.3, pp.356-370, 2011.
- [18]. N. Gayatri, "Performance analysis of data mining algorithms for software quality prediction", Advances in Recent Technologies in Communication and Computing International Conference on. IEEE, India, pp.393-395, 2009.
- [19]. Y. Zhao, "Comparison of decision tree methods for finding active objects", Advances in Space Research, vol.12, issue.14, pp.1955-1959, 2008.
- [20]. I. Gondra, "Applying machine learning to software fault-proneness prediction", Journal of Systems and Software, vol.81, issue.2, pp.186-195, 2008.
- [21]. M. Hall, "A decision tree-based attribute weighting filter for naïve Bayes", Knowledge-Based Systems, vol.20, issue.2, pp.120-126, 2007.
- [22]. C.J Du, "Learning techniques used in computer vision for food quality evaluation: a review", Journal of food engineering, vol.72, issue.1, pp.39-55, 2006.
- [23]. J. Huang, "Using AUC and accuracy in evaluating learning algorithms", IEEE Transactions on knowledge and Data Engineering, vol.17, issue.3, pp.299-310, 2005.
- [24]. M. R Boutell, "Learning multi-label scene classification", Pattern recognition, vol.37, issue.9, pp.1757-1771, 2004.
- [25]. J. Huang, "Comparing naïve Bayes, decision trees, and SVM with AUC and accuracy" Data Mining Third IEEE International Conference on IEEE, USA, pp.553-556, 2003.
- [26]. D. Zhang, "Question classification using support vector machines", Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, USA, pp.26-32 2003.
- [27]. D. Zhang, "Machine learning and software engineering", Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings. 14th IEEE International Conference on. IEEE, 2002.
- [28]. I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective", Artificial Intelligence in medicine vol. 23, issue.1, pp.89-109, 2001.
- [29]. A.K Jain, "Classification of text documents", Pattern Recognition, Fourteenth International Conference on IEEE, Australia, pp.1051-4651, 1998.

- [30]. A.P Bradley, "*The use of the area under the ROC curve in the evaluation of machine learning algorithms*" Pattern recognition, vol.30, issue.7, pp-1145-1159, 1997.
- [31]. S.W Wilson, "*Classifier fitness based on accuracy*", Evolutionary computation, vol.3, issue.2, pp.149-175, 1995.

### Authors Profile

---

Miss. Jyoti Devi pursuing masters in Computer Science and Engineering from Baddi University, himachal pradesh. She completed her bachelor's at vaishno college of Engineering for affiliated to Himachal Pradesh Technical University. She published a paper in IJCSMC journal. Her main research paper is support vector machine classifier.



Mrs. Nancy sehgal currently working as a Assistant Professor at Baddi University. She was mtech honor and has done many research in various domains. She has published many papers. Her main research paper focuses on image steganography.

