

Feature Selection Using Decision Tree Induction in Class level Metrics Dataset for Software Defect Predictions

N.Gayatri, S.Nickolas, A.V.Reddy

Abstract— The importance of software testing for quality assurance cannot be over emphasized. The estimation of quality factors is important for minimizing the cost and improving the effectiveness of the software testing process. One of the quality factors is fault proneness, for which unfortunately there is no generalized technique available to effectively identify fault proneness. Many researchers have concentrated on how to select software metrics that are likely to indicate fault proneness. At the same time dimensionality reduction (feature selection of software metrics) also plays a vital role for the effectiveness of the model or best quality model. Feature selection is important for a variety of reasons such as generalization, performance, computational efficiency and feature interpretability.

In this paper a new method for feature selection is proposed based on Decision Tree Induction. Relevant features are selected from the class level dataset based on decision tree classifiers used in the classification process. The attributes which form rules for the classifiers are taken as the relevant feature set or new feature set named Decision Tree Induction Rule based (DTIRB) feature set. Different classifiers are learned with this new data set obtained by decision tree induction process and achieved better performance. The performance of 18 classifiers is studied with the proposed method. Comparison is made with the Support Vector Machines (SVM) and RELIEF feature selection techniques. It is observed that the proposed method outperforms the other two for most of the classifiers considered. Overall improvement in classification process is also found with original feature set and reduced feature set. The proposed method has the advantage of easy interpretability and comprehensibility. Class level metrics dataset is used for evaluating the performance of the model. Receiver Operating Characteristics (ROC) and Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) error measures are used as the performance measures for checking effectiveness of the model.

Index Terms — Classification, Decision Tree Induction, Feature Selection, Software metrics, Software Quality, ROC.

I. INTRODUCTION

The demand for software quality estimation has been tremendously growing in recent years. As a consequence, issues related to the testing have become crucial [1]. The software quality assurance attributes are Reliability, Functionality, Fault Proneness, Reusability and Comprehensibility [2]. Among these defect prediction/fault

proneness is an important issue. It can be used in assessing the final product quality, estimating the standards and satisfaction of customers. Fault proneness can also be used for decision management with respect to the resource allocation for testing and verification. It is also one of the quality classification tasks of software design in which prediction of fault prone modules in the early design phase emphasizes the final quality outcome within estimated time and cost [3].

Variety of software defect prediction techniques are available and they include statistical, machine learning, parametric and mixed model techniques [4]. Recent studies show that many researchers used machine learning for software quality prediction. Classification and Clustering are some approaches in machine learning where classification is being used widely now [5][6]. For the effective defect prediction models, the data/features to be used also play an important role. If the data available, is noisy or features are irrelevant then prediction with that data results in inefficient outcome of the model. So the data must undergo the preprocessing so that data can be clean without noise and less redundant. One of the important steps in data preprocessing is feature selection [6]. Feature selection selects the relevant features i.e., irrelevant features are eliminated so as to improve the efficiency of the model. In literature, many feature selection techniques have been proposed [7].

In this paper, a decision rule induction method for feature selection is proposed. The features appeared in the rules when the classifier is learned with the decision tree classifiers, are formed as relevant features. These new features are given as input to other classifiers and the performances of the model using these reduced features are compared. This method is more comprehensible (easy to understand and interpret) when compared with others because the tree algorithms form rules which are understandable and easy to interpret.

The class level metrics dataset which is available from promise repository named KC1 is used here for defect predictions [8]. The data set contains 94 metrics and one class label i.e. defective or not defective from which relevant features are obtained. Different classifiers are used for the comparison of the proposed approach with other feature selection methods like Support Vector Machines (SVM) and RELIEF which are found as new methods for software predictions from the literature.

The performances of the classifiers are compared using Receiver Operating Characteristics (ROC) curve and error values like MAE and RMSE. Receiver Operating Characteristics analysis is a tool that realizes possible combinations of misclassifications costs and prior

N.Gayatri is with National Institute of Technology, Trichy working as Research Scholar in Computer Applications Department. (Email: gayatrinandam@yahoo.co.in)

S.Nickolas is with Department of Computer Applications working as Associate Professor at National Institute of Technology, Trichy., India (914312503739, Email: nickolas@nitt.edu)

A.V.Reddy is with Department of Computer Applications working as Professor at National Institute of Technology, Trichy.,India (Email: reddy@nitt.edu)

probabilities of fault prone (fp) and not fault prone (npf) [9]. ROC is taken as the performance measure because of its robustness towards imbalanced class distributions and to varying an asymmetric misclassification costs [10]. MAE and RMSE are the error measures and these values should be low for an effective model.

The paper is structured as follows: Section 2 gives the detailed related work, section 3 explains the proposed work and in section 4, experimental setup is discussed followed by a brief analysis of experimental results in section 5 and the results at the end.

II. RELATED WORK

Now a days machine learning is applied for software domain to classify the software modules as defective or not defective, so that early identification of defective modules can be corrected and tested before the final release for the module. This may lead to the quality outcome of the module and also there may be cost benefit.

Classification is a popular approach for software defect prediction and categorizes the software code attributes into defective or not defective, which is done by means of a classification model derived from software metrics data of previous development projects [11]. Various types of classifiers have been applied for this task including statistical methods [12], tree based methods [13][14], neural networks [15]. Data for defect prediction is available in large extent from the data sources [8].

One of the problems with large databases is high dimensionality, which means numerous unwanted and irrelevant data are present which causes erroneous, unexpected and redundant outcome of the model. Sometimes irrelevant features may lead to complexities in classification, so this irrelevant data must be eliminated so as to get the best outcome. Therefore data dimensionality reduction techniques such as feature selection or feature extraction have to be employed. Much research work has been done on this dimensionality reduction [16]. Many new Feature selection techniques have been proposed. The Feature selection is selecting features from wrapper or filter model i.e. we select from already existing or based on the ranking of the attributes or with the correlation between the variables and classes [16][17]. But Feature extraction is the generating components based on the data present. Additional components are generated which represent the overall dataset for which classification is done. Feature relevance and selection for classification has wide scope for research in recent years [18]. There are two categories of feature selection methods namely: Filters and Wrappers. Filter methods select the features by without constructing the predictive accuracy of the model, but by heuristically determined relevant knowledge [17], where as wrapper method chooses the relevant features based on the predictive accuracy of the model [19]. Research shows that wrapper model outperforms the filter model by comparing the predictive power on unseen data [20]. Wrapper method uses accuracy of the model on the training dataset as a measurement of how well a subset of features are formed and turns feature selection problem into optimization problem. On the other hand Filter feature selection techniques give the ranking of the features, where top ranked features are selected as best features [17]. Much research has been done in recent years and many have developed different feature selection techniques based on

different evaluation and searching criteria. Correlation based feature selection, Chi-square feature selection, Information gain based on entropy method, Support vector machine feature selection, Attribute Oriented Induction, Neural Network feature selection method, Relief feature selection method are some of feature selection methods available in the literature. These include filters and wrapper feature techniques. Some Statistical methods are also used for feature selection like Factor Analysis, Discriminant Analysis, and Principal Component Analysis etc. For all feature selection techniques different search criteria are applied. Some of the above feature techniques are also applied for software engineering domain for identifying relevant feature set which improves the performance of the model for defect identification.

III. PROPOSED WORK

In our Feature selection approach, a Decision tree induction is used for selecting relevant features. Decision tree induction is the learning of decision tree classifiers. It constructs a tree structure where each internal node (non leaf node) denotes the test on the attribute. Each branch represents the outcome of the test and each external node (leaf node) denotes the class prediction. At each node the algorithm chooses the best attribute to partition data into individual classes. The best attribute for partitioning is chosen by the attribute selection process with Information gain measure. The attribute with highest information gain is chosen for splitting the attribute. The information gain is of the attribute is found by

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

where p_i is the probability that a arbitrary vector in D belongs to class c_i . A log function to the base 2 is used, because the information is encoded in bits. Info (D) is just the average amount of information needed to identify the class label in vector D. Before constructing the trees base cases have to be taken in to consideration with following points:

- If all the samples belong to the same class, it simply creates the leaf node for the decision tree.
- If no features provide any information gain, it creates a decision node higher up the tree using the expected value of the class.

The algorithm for decision tree induction is given as follows

1. Check for base cases.
2. For each attribute a, find the information gain of each attribute for splitting
3. Let a-best be the attribute with highest information gain
4. Create a decision node that splits on a-best
5. Recur on the sub lists obtained by splitting on a-best , and add those nodes as children for the tree.

The trees are constructed from top down recursive approach which starts with training set of tuples and their associated class labels. The training set is recursively partitioned into smaller subsets as the tree is built. After the tree is built, for easy interpretation the rules are extracted using the leaf nodes of the tree, because rules give more comprehensibility than tree structure in case of big dataset.

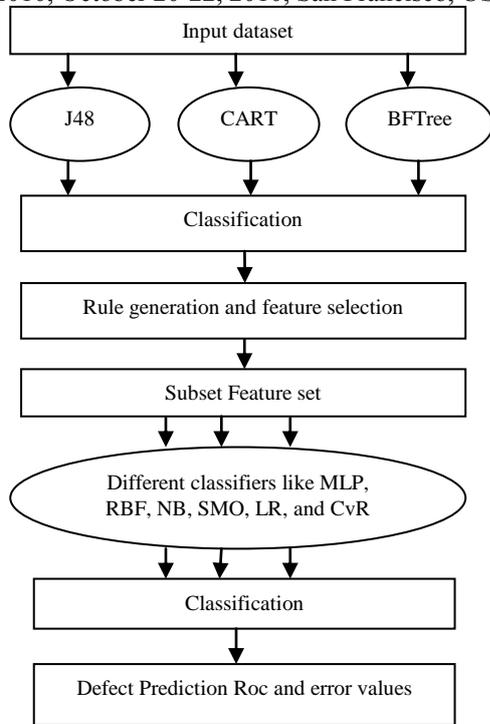


Fig 1: Proposed Architecture

To extract Rules from the trees, each path from the root to leaf node creates a rule, and each splitting criteria along the given path is logically ANDed to form the rule antecedent. The leaf node holds the class predictions, forming the rule consequent because the rules are extracted directly from the trees, they are mutually exclusive. The features which appeared in the rules are selected as the relevant features. All the other features which did not appear in the rules are considered as irrelevant.

In our approach we have used three decision tree algorithms given for classification for which the classification is done using decision tree induction and trees are constructed by rule generation using the input dataset. All the features which are found in the rules are selected collectively and they form the subset feature set. When this new feature set is learned with the same classifiers, the performance of the classifier is improved. The architecture of the proposed work is shown in Fig 1.

The algorithm has advantage of

1. Handling both continuous and discrete attributes
2. Handling training data with missing attribute values
3. Handling attributes with differing costs.
4. Pruning trees after creation

Using the proposed method, only 15 features out of 94 features are found as relevant features. So 80% of reduction is found. The frequency of features appeared in the rules are shown graphically in Fig 2. The features obtained from proposed feature selection method and the other feature selection techniques like Support Vector Machines and Relief are compared for performance evaluation. 18 classifiers are used for finding effectiveness of the proposed method.

IV. EXPERIMENTAL SETUP

A. Dataset description

The data set is the class level dataset named KC1 which contains class level metrics and method level metrics.

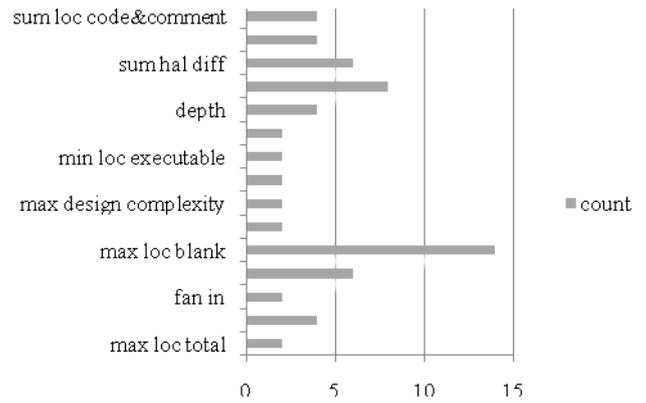


Fig 2: Frequency of the variables appeared in the rules

There are only four method level metrics. Koru et al. [20] converted method-level metrics into class-level ones using minimum, maximum, average and sum operations for KC1 dataset. 21 method-level metrics were converted into 84 class-level metrics. There were 84 metrics derived from transformation and 10 metrics from class-level metrics to create 94 metrics with 145 instances and one class attribute.

B. Description of feature selection Algorithms: RELIEF and SVM n of feature selection

We have used RELIEF and Support Vector Machine feature selection techniques used for comparison with the proposed method which is described below:

RELIEF [21] is one of the popular techniques found in the literature. The algorithm assigns weight to a particular features based on the difference between feature values of nearest neighbor pairs. Cao et.al further developed this method by learning feature weight in kernel spaces. RELIEF algorithm evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class which can operate on both discrete and continuous class data.

SVM evaluates or gives the feature based on the ranking of the attributes. It evaluates the worth of an attribute by using an SVM classifier. Attributes are ranked by the square of the weight assigned by the SVM. Attribute selection for multiclass problems is handled by ranking attributes for each class separately using a one-vs.-all method and then "dealing" from the top of each pile to give a final ranking[22]. For the experimentation we have used WEKA an open source data mining tool. All the classifiers and feature selection techniques are experimented using default parameters in WEKA [26].

C. Performance Measures

Different performance measures are available for model effectiveness. They are given below .

In a binary (positive and negative1) classification problem, there can be four possible outcomes of classifier prediction: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).

Table 1: Confusion matrix

	Correct result		
	+	-	
Obtained result	+	TP	FP
	-	FN	TN

A two-by-two confusion matrix is described in Table 1. The four values TP, TN, FP and FN provided by the confusion matrix form the basis for several other performance metrics that are well known and commonly used within the data mining and machine learning community, where N represents the number of instances in a given set. The Overall Accuracy (OA) provides a single value that ranges from 0 to 1. It can be calculated by the following equation

$$OA = \frac{TP + TN}{N}$$

where N represents the total number of instances in a data set. While the overall accuracy allows for easier comparisons of model performance, it is often not considered to be a reliable performance metric, especially in the presence of class imbalance [23].

Root Mean Squared Error (RMSE): The Mean-Squared Error is one of the most commonly used measures of success for numeric prediction. This value is computed by taking the average of the squared differences between each computed value (c_i) and its corresponding correct value (a_i). The Root Mean-Squared Error is simply the square root of the Mean-Squared Error. The Root Mean-Squared Error gives the error value the same dimensionality as the actual and predicted values.

$$\sqrt{\frac{(a_1 - c_1)^2 + (a_2 - c_2)^2 + \dots + (a_n - c_n)^2}{n}}$$

Mean Absolute Error (MAE): Mean Absolute Error is the average of the difference between predicted and actual value in all test cases; it is the average prediction error.

$$\frac{|a_1 - c_1| + |a_2 - c_2| + \dots + |a_n - c_n|}{n}$$

RMSE and MAE suggest that the error rate is very small, which can be considered as a measure of effectiveness of the model.

The **Area under curve (AUC)** i.e., Receiver Operating Characteristic curve (**ROC**) is a single-value measurement that originated from the field of signal detection. The value of the AUC ranges from 0 to 1. The ROC curve is used to characterize the trade-off between true positive rate and false positive rate. A classifier that provides a large area under the curve is preferable over a classifier with a smaller area under the curve. A perfect classifier provides an AUC that equals 1. The advantages of the ROC analysis are its robustness toward imbalanced class distributions and to varying and asymmetric misclassification costs [24]. Therefore, it is particularly well suited for software defect prediction tasks. In this work we

Table 2: ROC values for the classifiers

Learning method	Original feature set	SVM feature set	RELIEF feature set	DTRB feature set
J48	0.722	0.733	0.731	0.798
BFTree	0.733	0.719	0.726	0.703
Random forest	0.782	0.811	0.794	0.847
CART	0.72	0.706	0.731	0.772
Naïve Bayes	0.789	0.844	0.827	0.846
Logistic regression	0.718	0.854	0.845	0.826
Multi layer Perceptron	0.733	0.845	0.709	0.828
RBF	0.771	0.823	0.827	0.814
SMO	0.798	0.794	0.718	0.792
IBK	0.718	0.687	0.659	0.739
KStar	0.73	0.793	0.754	0.805
CvR	0.789	0.84	0.824	0.84
Ensemble	0.821	0.82	0.811	0.836
VFI	0.757	0.801	0.77	0.818
DTNB	0.773	0.735	0.774	0.742
JRIP	0.73	0.743	0.736	0.789
PART	0.708	0.742	0.734	0.755
Conjunctive rule	0.733	0.739	0.742	0.739

and RMSE error measures as the performance measures as they have been used widely better than Accuracy and other measures for performance evaluation.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The results obtained with new feature set and the KC1 dataset are compared with the two feature selection techniques SVM and RELIEF. The comparison between the original feature set with all 94 attributes, and the reduced new feature set using proposed method is done. 18 classifiers are used for the defect prediction with cross validation. Cross-validation (CV) tests exist in a number of ways but the general idea is to divide the training data into a number of partitions or folds. The classifier is evaluated by its classification accuracy on one partition after having learned from the other. This procedure is then repeated until all partitions have been used for evaluation [25]. Some of the most common types are 10-fold, n-fold and bootstrap. The difference between these three types of CV lies in the way data is partitioned. 10 fold cross validation is used for evaluation, which is one of the most widely used and acceptable methods for evaluating machine learning techniques [25].

A. Performance of Classifiers with new feature set using ROC

From the Table 2 it is observed that Random forest and Naïve Bayes algorithms whose ROC=0.847 out performs all the other algorithms with the new approach. Defect prediction with this feature selection algorithms give better classification of fault prone and not fault prone modules of metrics dataset, when compared to others, Ensemble algorithm achieved slightly better ROC. The ROC value is 0.836. VFI (voting feature interval) also has ROC as 0.818. So RF, NB, Ensemble, VFI algorithms are effective for classification of software defects using proposed method. Classifications via Regression (CvR) are used. From these,

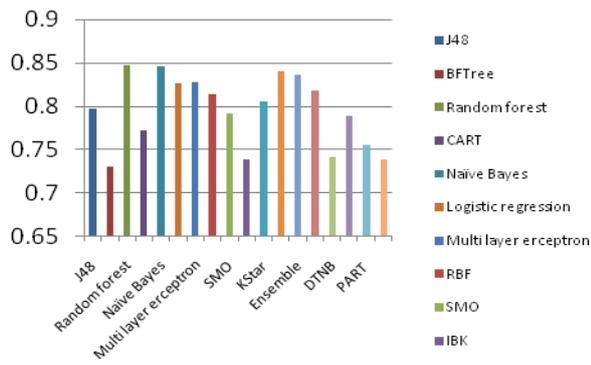


Fig 3: Ranking of Classifiers using proposed feature selection method

CvR achieves better roc than the other two, next comes logistic regression and last is CART. Generally regression learning problems lead to poor numeric estimates but here they can be used for defect prediction. It is observed that MLP achieves better performance over RBF for Neural Network techniques. SMO is support vector classifiers used for prediction, whose performance is less comparatively with the new method when compared to other classifiers, but it can be comparable. The other classifiers have less ROC comparatively with the new approach; so these classifiers are also preferred for defect predictions. The ranking of classifiers for the proposed approach is shown in Fig 3.

B. Performance of Classifiers when MAE and RMSE error measures are taken into consideration:

Table 3 gives the MAE and RMSE values for original and reduced feature sets. These values are depicted graphically in Fig 4 and 5. From the Fig 4&5, it is observed that for all the classifiers the error values are reduced with new(DTIRB) feature set (RMSE) when compared to original values (RMSEO), except for CART and MLP. For these there is slight increase in MAE and RMSE. So using MAE and RMSE, these algorithms may be less preferable for defect predictions. Other than that for all the classifiers, new feature selection method gives better results.

C. Analysis of 18 classifiers with three feature selection techniques

For the two feature selection techniques SVM and RELIEF methods, the feature selection is done based on ranking. The top 15 attributes are selected for classification of models. In new feature set also only 15 attributes appear in the rules. So reduced (DTIRB) feature set has 15 features. From the Fig 6, it is observed that MLP and RBF achieves better results with SVM feature selection when compared to proposed method and RELIEF feature selection method. So for NN algorithms SVM feature selection technique may be preferred. SMO achieves better and consistent result with SVM feature selection method and proposed method than RELIEF method. So SMO may not be preferable for defect prediction using RELIEF.

The conjunctive rule algorithm which comes under rules category in WEKA gives consistent result with SVM and proposed method and slightly better result with RELIEF algorithm. Other than the above algorithms, all others perform better with the proposed method in terms of ROC. So, from the results it is observed that the observed that the ROC values for the proposed method are high and error

Table 3: Error values for original dataset and reduced dataset

Learning method	Full feature set		Reduced feature set Using DTIRB method	
	MAE	RMS E	MAE	RMSE
J48	0.31	0.52	0.27	0.47
BFTree	0.30	0.50	0.28	0.45
Random Forest	0.32	0.43	0.27	0.39
CART	0.34	0.46	0.37	0.45
Naive Bayes	0.31	0.55	0.25	0.48
Logistic regression	0.31	0.56	0.27	0.40
RBF	0.34	0.44	0.33	0.42
Multi layer Perceptron	0.40	0.28	0.23	0.43
SMO	0.2	0.44	0.20	0.45
IBK	0.29	0.54	0.29	0.54
Kstar	0.31	0.53	0.28	0.49
CvR	0.32	0.43	0.31	0.41
VFI	0.46	0.49	0.43	0.45
Ensemble	0.35	0.40	0.34	0.40
DTNB	0.46	0.30	0.34	0.42
JRip	0.35	0.45	0.31	0.45
PART	0.31	0.54	0.29	0.47
Conjunctive Table	0.44	0.33	0.33	0.42

values are low for most of the classifiers, i.e. the proposed method achieves better performance for software defect predictions and it can be used widely for software defect predictions.

VI. CONCLUSION

Performances of learning algorithms may vary using different classifiers, different performance measures and different feature selection methods. The selection of appropriate classification algorithm and feature selection method is an important task.

In this paper, a feature selection method based on decision rule induction for software defect prediction is proposed. Selection of the relevant features is done by using the rules of the decision tree classifiers. Out of 94 features only 15 features are selected using the proposed method.

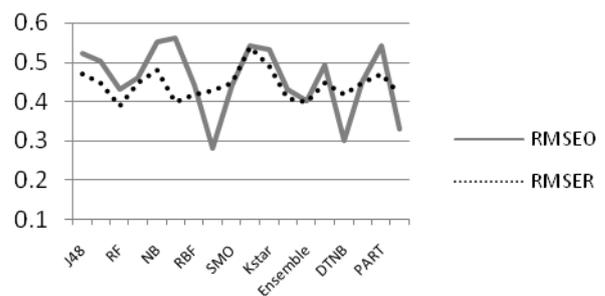


Fig 4: Performance of new feature set using RMSE

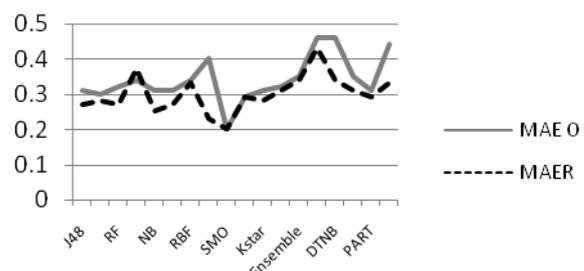


Fig 5: Performance of new feature set using MAE

Classification built on this new feature set has significant differences in performance when compared with complete set of features for defect predictions. This would benefit the metrics collection, model validation and model evaluation time of future software project development efforts of similar systems. The other two feature selection techniques, namely RELIEF and SVM are used and compared with the proposed method. The new approach resulted in better performance comparatively in terms of ROC and Error measures. So the new method can be used widely for software defect predictions. The proposed method is more comprehensible than others and easily interpretable. The performance measures taken here is ROC and Error measures which are found to be the best measures for software defect predictions. The future scope will be comparing many machine learning techniques and statistical feature selection techniques with the proposed approach for different dataset and various other performance measures.

REFERENCES

[1] Iker Gondra, "Applying machine learning to software fault-proneness prediction", The journal of System and Software, Pg 186-195,2008.
[2] N.E. Fenton and S.L Pfleeger, "Software Metrics, A Rigorous & Practical Approach", International Thomson Computer Press, London, 1997.
[3] Raimund Moser, Witold Pedrycz, Giancarlo Succi, "A Comparative Analysis of the Efficiency of Change Metrics and Static Code Attributes for Defect Prediction", ICSE'08 ,PP 181-190,May 10-18,2008,Germany.
[4] Venkata U.B.Challagulla,Farokh B, I-Ling Yen,Raymond A.Paul,"Emperical Assessment of Machine Learning based Software Defect Prediction Techniques", Proceedings of the 10th International Work Shop on Object -Oriented metrics.
[5] Quinlan, J. R., "C4.5: Programs for Machine Learning", SanMateo, CA: Morgan Kaufmann Publishers, 1993.
[6] Han, J., & Kamber, M., "Data Mining: Concepts and Techniques", San Francisco: Morgan Kaufmann Publishers,2001
[7] Almuallim,H., and Dietterich,T.G., "Efficient algorithms for identifying relevant features" In Proceedings of Ninth Canadian Conference on Artificial Intelligence, Vancouver, BC: Morgan Kaufmann, 1992.
[8] Promise Software Engineering, <http://promise.site.uttowa.ca/SERpository>
[9] Stefan Lessmann, , Bart Baesens, Christophe Mues, and Swantje Pietsch. S. Lessmann and S. Pietsch, " Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and

Novel Findings", IEEE Transactions On Software Engineering, Vol. 34, No. 4, July/August 2008 .pp 485-496.
[10] F. Provost and T. Fawcett, "Robust Classification for Imprecise Environments," Machine Learning, vol. 42, no. 3, pp. 203-231, 2001
[11] N.F. Schneidewind, "Methodology for Validating Software Metrics," IEEE Trans. Software Eng., vol. 18, no. 5, pp. 410-422, May 1992
[12] T.M. Khoshgoftaar and E.B. Allen, "Logistic Regression Modeling of Software Quality," Int'l J. Reliability, Quality and Safety Eng. vol. 6, no. 4, pp. 303-317, 1999
[13] L. Guo, Y. Ma, B. Cukic, and H. Singh, "Robust Prediction of Fault-Proneness by Random Forests," Proc. 15th Int'l Symp. Software Reliability Eng., 2004.
[14] T.M. Khoshgoftaar, E.B. Allen, W.D. Jones, and J.P. Hudepohl, "Classification-Tree Models of Software-Quality over Multiple Releases," IEEE Trans. Reliability, vol. 49, no. 1, pp. 4-11, 2000.
[15] M.M. Thwin, T. Quah, Application of neural networks for software quality prediction using object-oriented metrics, in: Proceedings of the 19th International Conference on Software Maintenance, Amsterdam, The Netherlands, 2003, pp. 113-122
[16] Almuallim,H., and Dietterich,T.G., "Efficient algorithms for identifying relevant features" In Proceedings of Ninth Canadian Conference on Artificial Intelligence, Vancouver, BC: Morgan Kaufmann, 1992
[17] Ooi,C.H., Chetty,M., & Teng,S.W.: Differential prioritization in feature selection and classifier aggregation for multiclass microarray datasets", Data mining and Knowledge Discovery, pp 329-366, 2007
[18] Hall, M. A., & Holmes, G., "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining", IEEE Transactions on Knowledge and Data Engineering, 15, pp 1437-1447, 2003.
[19] G.H. John, R. Kohavi, K. Pfleger, "Irrelevant Features and Subset Selection Problem", Proceedings of the Eleventh International Conference of Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA (1994 121-129).-129.
[20] A.G. Koru, H. Liu, "An investigation of the effect of module size on defect prediction using static measures", in: Workshop on Predictor Models in Software Engineering, St. Louis, Missouri, 2005, pp. 1-5.
[21] Marko Robnik-Sikonja, Igor Kononenko: An adaptation of Relief for attribute estimation in regression. In: Fourteenth International Conference on Machine Learning, 296-304, 1997.
[22] I. Guyon, J. Weston, S. Barnhill, V. Vapnik (2002). Gene selection for cancer classification using support vector machines. Machine Learning. 46:389-422.
[23] R. Arbel and L. Rokach. Classifier evaluation under limited resources. *Pattern Recognition Letters*, 7(14):1619-1631, 2006
[24] F. Provost and T. Fawcett, "Robust Classification for Imprecise Environments," Machine Learning, vol. 42, no. 3, pp. 203-231, 2001.
[25] N. Laves son and P. Davidson, "Multi-dimensional measures function for classifier performance", 2nd. IEEE International conference on Intelligent system, pp.508-513, 2004
[26] WEKA: <http://www.cs.waikato.ac.nz/ml/weka>.

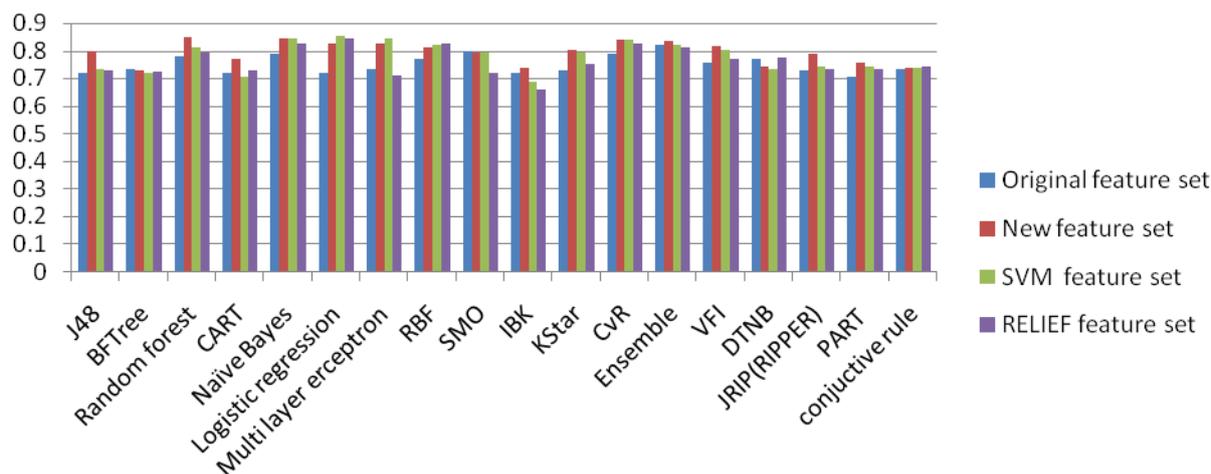


Fig6: Performance comparison of three features selection methods in terms of ROC