# Improved Random Forest Algorithm for Software Defect Prediction through Data Mining Techniques

Kalai Magal. R
ME- Software Engineering,
Department of Computer Science and Engineering,
SSN College of Engineering ,Old Mahabalipuram Road,
Kalavakkam – 603 110,Tamil Nadu, India.

Shomona Gracia Jacob,
Associate Professor,
Department of Computer Science and Engineering,
SSN College of Engineering,Old Mahabalipuram Road,
Kalavakkam – 603 110,Tamil Nadu, India

## ABSTRACT

Software defect prediction using classification algorithms was advocated by many researchers.Moreover the classifier ensemble can effectively improve classification performance compared to a single classifier. The research on defect prediction using classifier ensemble methods are motivated since they have not been fully exploited.Software defects leads to failure of many defense systems. A comparative study of various classification methods was performed to classify software defects. The methods include Random Tree, Random Forest, Bayesian Network, Naive Bayes, K-Nearest Neighbour and Instance Based Classifier.Random Forest algorithm was found to give more accurate prediction than other classifiers.

To enhance the classification accuracy the new algorithm ”Improved Random Forest” is proposed. It works by incorporating best feature selection algorithm with the Random Forest to gives better accurracy. Correlation based Feature Subset Selection algorithm selects the optimal subset of features. The optimal features are fed as a part of Random Forest classification to give better accuracy in software defect prediction. The six optimal subset of features were selected for PC1 dataset. The features are selected by the CFS and utilized by Random Forest to improve the accuracy of existing Random Forest. The experiments were carried on public-NASA datasets of PROMISE repository.

## Keywords

Software Defect Prediction, Feature Selection, Classification, Classifier Evaluation.

## 1. INTRODUCTION

Data mining is the task of investigating data from various perspectives and organizing the data into relevant and meaningful information**[1]**. There are numerous data mining algorithms such as classification, regression, association, clustering, *etc*,. used in software quality analysis. This paper uses Feature Selection and classification approach for the prediction of defective software **[2]**. Feature selection is the method of deciding on a subset of important and relevant features for building reliable learning models. It makes training and utilizing a classifier more efficient by reducing the size of the effective training set. Moreover feature selection often increases classification accuracy by removing noisy features. Classification approach divides the data samples into target classes. For example, software module can be categorized into “defective” or “non-defective” using classification approaches. Defect in a software module occurs due to source code error that further produces wrong output and leads to poor quality software products. Defective software modules are also responsible for high development and maintenance cost and customer dissatisfaction. The NASA Space Network, also referred to as the Tracking and Data Relay Satellite System consists of nine on-orbit telecommunications satellites stationed at geo-synchronous stationary positions.

In this paper, we apply classification algorithms on publicly available datasets of the NASA PROMISE repository in order to classify the software modules as defective/non-defective. The datasets employed for this research were PC1 , PC2, PC3 and PC4 **[3].** This paper proposes a computational framework using data mining techniques to detect the existence of defects in software components. The framework comprises of feature selection, data classification and classifier evaluation. Correlation based feature subset selection, a feature-subset selection technique **[4]**, is used to determine the significant features that are prominently affecting the defect prediction in software modules. The efficiency of predictive model could be enhanced with reduced feature set obtained after feature selection and further used to identify defective modules in a given set of inputs. This paper evaluates the performance of the proposed model. The experimental results indicate the effectiveness of the proposed feature selection based predictive model based on standard performance evaluation parameters

## 2. LITERATURE SURVEY

The work carried out thus far by other researchers that are related to defect prediction research using feature reduction and classification is concisely presented here.In the survey of Bhekisipho Twala**[7]** the performance of different ensembles was compared for software fault prediction. The results for each classifier was used as a baseline.The correlation maximisation method was used to select the appropriate number of ensemble classifier members, of which three classifiers per ensemble were chosen. For each ensemble,four sampling procedures (bagging, boosting, feature selection,and randomization) were considered. This was the case for each individual datasets.To empirically evaluate the performance of one of the top five classifiers in data mining (AR, DT, k-NN, NBC and SVM), an experiment was conducted on four datasets in terms of misclassification error rate. For each dataset, different types of metrics were used to predict models that were likely to predict faults. Three of the four datasets were collected by the NASA metrics data program (MDP) data repository.Three projects (CM1, JM1 and PC1) were comprised of which only partial requirement metrics were available. There were 10 attributes that described the requirements.A motivation for ensemble was the combination of outputs of many weak classifiers produces powerful ensembles with higher accuracy than a single classifier obtained from the same sample.Thus the ensemble classifier should be improved to give better results. In the survey of

Sonali Agarwal et al **[11]** feature selection based LSTSVM model for defect prediction was proposed. F-score feature selection technique was used to select significant features which are helpful to predict defective software modules.F-score is one of the simple and significant feature selection technique which is mostly used in machine learning. It calculates the discrimination between two sets of real numbers.The larger value of F-score indicates that the corresponding feature is more discriminative or highly significant.A disadvantage of F-score was that it does not reveal mutual information among features.The Fscore is a ratio of two variables: F = F1/F2, where F1 is the variability between groups and F2 is the variability within each group. In other words, a high F value means that at least one of the groups is significantly different from the rest, but it doesn’t tell which group.In order to address this issue,CFS is used which gives the optimal subset of features and with mutual information. In the survey of Catal and Diri et al(2009),**[13]** the impact of Random Forests was studied and algorithms based on artificial immune systems was used.It was analyzed that the area under the Receiver Operating Characteristics (ROC) curve was used as a performance evaluation measure. NASA datasets were utilised for this task. Their results showed that Random Forest achieved the highest accuracy rates than other methods such as NBC for small datasets. It did not performed well for large datasets.Thus, Random Forest algorithm should be improved in this project to give better accuracy for both small and large datasets.

## 3. PROPOSED METHODOLOGY

The proposed methodology is diagrammatically presented in Figure 1. The methodology involves data collection, feature selection, classification and performance evaluation. The NASA datasets of PROMISE repository namely PC1, PC2, PC3, PC4 were selected to classify models for software defect prediction. The datasets were collected and fed as input to the feature selection process. Correlation based feature subset selection method(CF subset) gave the optimal feature sets. It is an efficient feature selection algorithm, which gives high scores to subsets that includes features that are highly correlated to the class attribute, but have low correlation to each other. CFS evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. The selected features were used to classify into two classes namely defective and non- defective by using classification algorithms namely Bayes Net, Naive bayes, Random Forest, Instance based classifier and Random Tree. This was done to achieve the goal of being able to use the model to categorize the software as defective and Non defective. Random Forest yielded the highest accuracy with the reduced feature subset and is described below. Random Forest is a powerful new approach to data exploration, data analysis and predictive modeling. It performs error detection, generation of strong predictive models, etc . Random Forest algorithm will select the small subset of available attributes at random. It splits the node with the best variable among the available features. The embedded classifier Improved Random Forest is formulated to enhance the accuracy of the existing classifier.This is done by incorporating Correlation based

Feature subset selection algorithm and Random Forest algorithm The performance evaluation is carried out to evaluate and distinguish classes namely defectives and non defectives.

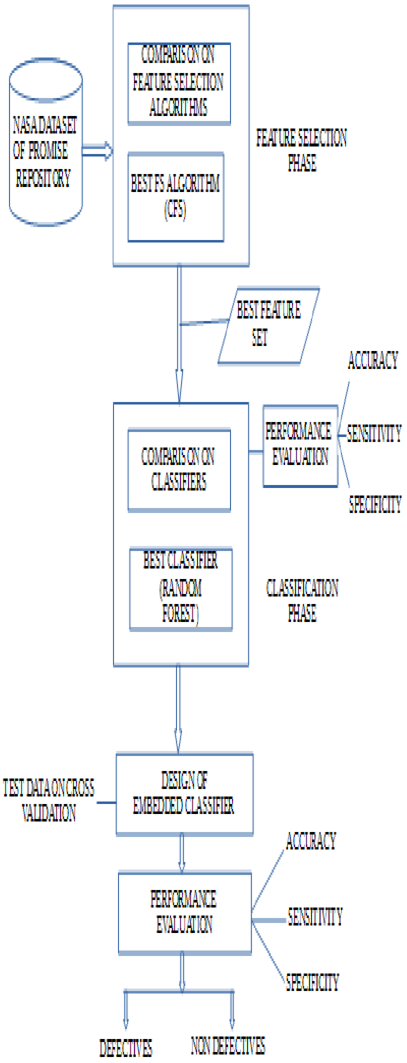

**Figure 1: Proposed Framework**

This is done by evaluating the performance measures such as accuracy, sensitivity and specificity. The performance evaluation is carried out with 10 fold cross validation data to give better accuracy. This in turn, produces a software defect prediction with better performance.

## 4. NASA DATASETS DESCRIPTION

Table 1 shows the NASA datasets description. It shows the attributes, instances, defects and non-defects count of PC1, PC2, PC3, PC4 NASA datasets [3]. The dataset is loaded into the Weka tool for furthur processing. The attribute, defect, instance and non-defect counts is different for different datasets.

**Table 1: NASA Datasets Description**

| Dataset | Attribute | Instance | Defects | Nondefect |
|---|---|---|---|---|
| PC1 | 22 | 1107 | 76 | 1031 |
| PC2 | 37 | 5460 | 23 | 5437 |
| PC3 | 38 | 1563 | 160 | 1403 |
| PC4 | 38 | 1399 | 178 | 1221 |

## 5 . COMPARISON ON DIFFERENT FS TECHNIQUES IN WEKA TOOL

**Table 2: Comparison on different FS techniques**

| Dataset | Attribute | CFS | Info Gain | GainRatio |
|---|---|---|---|---|
| PC1 | 22 | 6 | 22 | 22 |
| PC2 | 37 | 4 | 36 | 36 |
| PC3 | 38 | 6 | 37 | 37 |
| PC4 | 38 | 4 | 37 | 37 |

Table 2 shows the comparison of different feature selection techniques in the Weka tool. The feature selection algorithms like CFS,Gain Ratio and Information Gain were compared.The Gain Ratio and Information Gain are the ranking algorithms. It selects the attributes based on its ranks.CFS is the Correlation based feature subset selection algorithm. It gives the otimal subset of features. It gives high accuracy in selecting optimal features. Thus CFS is used to design the proposed algorithm.The CFS selects the minimum nuumber of attributes for all the datasets.For PC1 CFS selects six attributes,CFS selects four attributes for PC2,CFS selects six attributes for PC3 and CFS selects four attributes for PC4.

## 6. COMPARISON ON DIFFERENT CLASSIFICATION TECHNIQUES IN WEKA TOOL

**Table 3: Comparison on different classification techniques in Weka tool**

| Dataset | BN* | NB | IBK | RT | RF |
|---|---|---|---|---|---|
| PC1 | 72.4% | 89.1% | 92.1% | 91.6% | 92.9% |
| PC2 | 89.9% | 97.2% | 98.2% | 94.2% | 98.3% |
| PC3 | 70.2% | 48.7% | 87.5% | 87.6% | 89.9% |
| PC4 | 73.8% | 86.3% | 86.7% | 87.5% | 88.2% |

*BN-Bayesian Network, NB-Naive Bayes, IBK-Instance Based Classifier, RT-Random Tree, RF-Random Forest.

The Table 3 shows the performance accuracy of the classifiers compared with one another. The table shows that the Random Forest gives better results compared to other classifiers for all the datasets. The Table 3 clearly shows which classifier has higher accuracy. Though Random Tree and IBK gives high accuracy for some datasets, it does not performs well for other datasets when compared to Random Forest classifier.

Random Forest algorithm is chosen for the proposed work. The comparative performance evaluation of the classification algorithms is graphically presented in Figure 2. It is clear from the above results the Random Forest classifier has performed well in terms of accuracy. The accuracy of Random Forest classifier for PC1 data is 92.9%. For PC2 data the Random Forest classifier performance in terms of accuracy is 98.3%. For PC3 data Random Forest classifier performance in terms of accuracy is 89.9%. For PC4 data Random Forest classifier performance in terms of accuracy is 88.2%. Even though, other classifiers like instance based classifier and support vector machine gives high accuracy for some data, the Random Forest classifier yielded overall better performance.

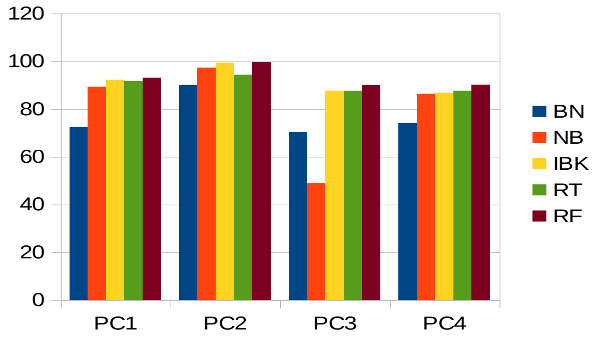

**Figure 2: Comparison of Classification Algorithms on Accuracy**

## 7. DESIGN OF EMBEDDED CLASSIFIER

The Embedded classifier is designed by adding new algorithm in the weka interfaced with netbeans.The added algorithm is implemented by incorporating Correlation based Feature subset selection algorithm with Random Forest algorithm to give better accuracy.The performance evaluation is done by cross validation technique to obtain better accuracy.The CFS selects the optimal subset of features and passes to the Random Forest thus it takes the optimal subset of features and gives better accuracy in classifying the defective and non defective software.

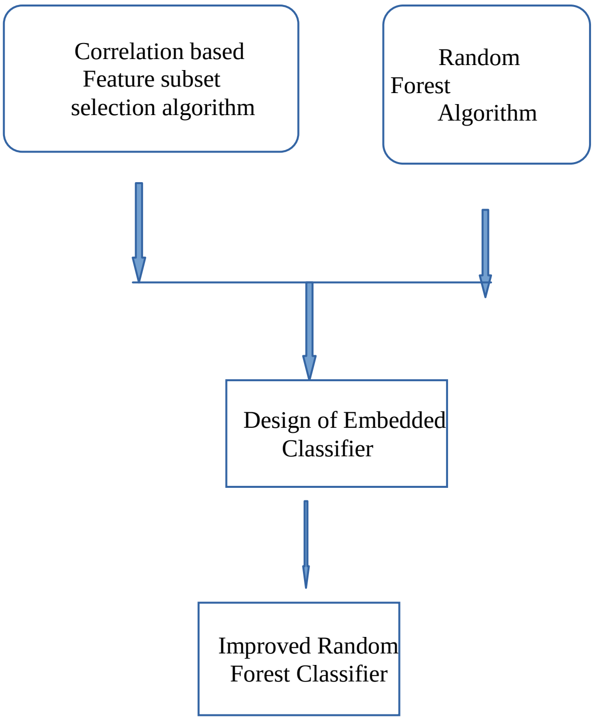
Correlation based Feature subset selection algorithm
Random Forest Algorithm
Design of Embedded Classifier
Improved Random Forest Classifier

**Figure 3: Design of Embedded Classifier**

## 5. RESULTS AND DISCUSSION

### 5.1 Comparison of Random Forest with Improved Random Forest

**Table 4: Comparison on different FS techniques in Weka tool**

| Datasets | RF* | IRF |
|---|---|---|
| PC1 | 92.953% | 94.545% |
| PC2 | 98.346% | 98.561% |
| PC3 | 89.208% | 89.676% |
| PC4 | 88.267% | 90.625% |

*RF-Random Forest, CFS-Correlation based Feature Subset Selection, IRF-Improved Random Forest

Table 4 shows the comparison of Random Forest and Embedded classifier Improved Random Forest classifier based on accuracy. The performance of the Improved Random Forest is high compared to the Random Forest classifier.

Figure 4 shows the graphical representation of the Imroved Random Forest compared with Random Forest. This shows that the Improved Random Forest gives better accuracy than Random Forest. This is achieved by evaluating the performance of the classifier in terms of accuracy.

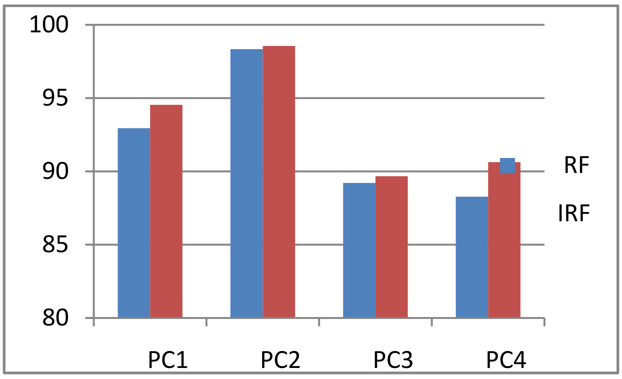

**Figure 4: Graphical Representation of Improved Random Forest**

The performance of proposed model was measured in terms of performance measures namely accuracy and specificity which further utilized for performance evaluation of the proposed model.

### *5.1.1. Accuracy*

Accuracy is also referred to as “correct classification rate” and is measured by taking the ratio of correct predictions to the total prediction made by the software defect prediction model and is formulated as:

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

### *5.1.2. Sensitivity*

Sensitivity, also called true positive rate, is estimated by calculating the percentage of correctly identified not-defective software modules and is formulated as:

Sensitivity= TP/ (TP+FN)

### *5.1.3. Specificity*

Specificity, also termed as true negative rate, is measured by calculating the percentage of correctly recognized defective modules and is formulated as:

Specificity= TN/ (TN+FP)

Where TP denotes True Positives, FP- False Positives, TN-True Negatives, FN-False Negatives respectively. The following tables detail the performance of five different classifiers on four NASA datasets PC1, PC2, PC3 and PC4.

It is clear from the above results the Improved Random Forest classifier has performed well in terms of accuracy. The accuracy of Improved Random Forest classifier for PC1 data is **94.5%.** For PC2 data the Improved Random Forest classifier performance in terms of accuracy is **98.5%.** For PC3 data the Improved Random Forest classifier performance in terms of accuracy is **89.6%** and for PC4 data is **90.6%.**

## 6. CONCLUSION

This study proposed a feature selection based Random Forest model for software defect prediction. Correlation based feature subset selection technique was used to select significant features which were helpful to predict defects in software modules. There was a significant difference in classifier’s performance that was developed using new feature subset as compared to the classifier built on complete feature set. This study has evaluated the predicting performance of proposed model for defective software modules and also performed a comparative analysis against five statistical and machine learning approaches using four PROMISE datasets. The experimental results revealed that the predictive capability of the proposed approach is better or at least comparable with other approaches. This research discloses the effectiveness of Correlation based feature subset selection (CF subset) based Random Forest approach in predicting defective software modules and suggests that the proposed model can be useful in predicting defective software based on the important attributes. In order to improve the accuracy and quality of software development, data mining techniques are used to analyze and predict large number of defect data collected in the software development.In order to improve the accuracy the existing algorithm is improved by incorporating the suitable feature selection algorithm as a part of Random Forest algorithm which gives better accuracy.Thus the accuracy of Embedded classifier Improved Random Forest is improved. The performance of Improved Random Forest obtained from the result for PC1 dataset is **94.545%**, PC2 dataset is **98.561%**, PC3 dataset is **89.676%** and for PC4 dataset is **90.625%**. The future enhancement will be design of an Improved Random Forest as a system and utilizing the same to predict potential targets in other areas of research such as Healthcare and Security threats.

## 7. REFERENCES

[1] Dr.R.Geetha Ramani, S.Vinodh Kumar, Shomona Gracia Jacob,”Predicting Fault Prone Software Modules Using Feature Selection and Classification through Data Mining Algorithms”,2012.

[2] J. Han and M. Kamber, ―Data Mining; Concepts and Techniques, Morgan Kaufmann Publishers, 2000.

[3] Software Defect Dataset, PROMISE REPOSITORY, http://promise.site.uottawa.ca/SERepository/datasets-page.html, **(2013)** December 4.

[4] Y. W. Chen and C. J. Lin, “Combining SVMs with various feature selection strategies”,

[5] Hassan Najadat and Izzat Alsmadi,”Enhance Rule Based Detection for Software Fault Prone Modules”, International Journal of Software Engineering and Its Applications,Vol. 6, No. 1, January, 2012

[6] Kehan Gao, Taghi M. Khoshgoftaar2, Huanjing Wang and Naeem Seliya,”Choosing software metrics for defect prediction: an investigation on feature selection techniques”,software – practice and experience,2011

[7] Twala B (2011) Predicting software faults in large space systems using machine learning techniques

[8] Breiman L. (2001). Random Forests. Machine learning, 45(1):5–32.

[9] L. Guo, Y. Ma, B. Cukic and H. Singh, “Robust prediction of fault proneness by random forests”, Proceedings of the 15th International Symposium on Software Reliability Engineering (ISSRE’04), **(2004)**, pp. 417–428.

[10] Geetha Ramani R, Shomona Gracia Jacob., “Improved Classification of Lung Cancer Tumors Based on Structural and Physicochemical Properties of Proteins Using Data Mining Models”, PLoS ONE (Impact Factor: 4.537) 8(3): e58772, 2013, ISSN: 1932-6203.

[11] Sonali Agarwal S.A. and Divya Tomar D.T. (2014). A feature selection based model for software defect prediction. International Journal of Advanced Science and Technology, 35:39–58..

[12] J. Kaur and Pallavi, “Data Mining Techniques for Software Defect Prediction”, International Journal of Software and Web Sciences (IJSWS), (**2013)**, pp. 54-57.

[13] Catal C. and Diri B. (2009). A systematic review of software fault prediction studies. Expert systems with applications, 36(4):7346–7354.

| S.NO | REFEREES COMMENTS |
|---|---|
| 1 | Some of text parts in the paper are taken from various other sources. These sections must be rewritten in authors own languages with proper citation to avoid any plagiarism cases. Authors should present the research analysis using their own interpretation and language rather compiling of the texts which is available elsewhere. Any presence instance of plagiarism may hamper the author’s credibility and reputation. |
| 2 | More comprehensive evaluations are needed. Authors should present the<br>Experimental results and its corresponding analysis in detail supported by graphical and tabular data. |
| 3 | All the listed references must be properly cited in the body of the paper to avoid any copyright issues. |
| 4 | Conclusion needs to be elaborated mentioning the future scope of the idea. |
| 5 | The presentation of the paper needs improvement. Different fonts and sizes are used throughout the paper. The paper does not comply with the IJCA paper template. |

| S.NO | AUTHORS RESPONSE |
|---|---|
| 1 | The authors have rewritten the paper and have been citated properly. |
| 2 | The authors have presented more comprehensive evaluations and experimental results that are mentioned in comments. |
| 3 | The authors have properly cited the listed references in the body of the paper. |
| 4 | The authors have elaborated the conclusion and mentioned the future scope of the idea. |
| 5 | The authors have improved the presentation of the paper with different fonts and sizes and complied with the IJCA paper template. |