International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-05, Issue-01, April 2019

Investigating the Accuracy of Software Defect Prediction using an Adaptive Computational Intelligence Approach

[1] M.Chalapathi Rao, [2]Dr.P.Suryanarayana Babu
[1]Research Scholar, [2]Research Supervisor, Department of Computer Science
Rayalaseema University Kurnool, Andhra Pradesh, India.

Abstract: Prediction of software defects in software development and maintenance process is an important issue in which with the apprehension of the overall success of software. Predicting software defects in the earlier phases not only improves the software quality, reliability, efficiency but also greatly reduces the software cost. However developing a vigorous model for detecting defects is a tough task and the literature has proposed many techniques. Open source projects such as Eclipse and Firefox have repositories of open source defects. These repositories are reported by the user defects. Users of these repositories are generally non-technical and cannot assign these defects to the correct class. A Trialing defect to developers, fixing them is a tedious and time-consuming task. This paper presents an adaptive Computational Intelligence approach called Relevance based Fuzzy C-Means (RFCM) clustering algorithm is proposed to predict and classify software defects into defective and non-defective modules. The proposed process is constructed by integrating Relevance Based Learning (RBL) and Fuzzy C-Means clustering techniques to retrieve useful information after the classification and enable them to group from different data perspectives. The Relevance based Fuzzy C Means (RFCM) algorithm which efficiently classifies and predicts the accuracy of software defect detection. This algorithm also makes use a heuristic feature selection using fitness function. The empirical analysis showed that RFCM can be used effectively with high accuracy rate. Furthermore, a comparison accuracy measure is applied to compare the proposed prediction model with the existing state of the art of the algorithms. The collected results showed that the RFCM attained better performance with respect to accuracy measures.

Keywords — Software bug prediction, prediction model, machine learning, Relevance Based Learning (RBL), Fuzzy C-Means, Relevance based Fuzzy C Means (RFCM).

I. INTRODUCTION

The development in the software technology has increased many software products. Maintenance of these software products and has become very tedious and challenging task. It is due to software life cycle contains maintenance activities. As software systems become more complex, the likelihood of having faulty modules in software. The subsistence of software defects influence the reliability, quality and cost of maintenance of software dramatically. It is also difficult to achieve defect-free software, because hidden defects occur most of the time. In addition, a real challenge in software engineering is increasing software defect prediction model that could predict the defective modules in the in the early hours phase [1]. Before it is delivered to customers, it is very important to predict and fix the defects because assuring software quality is a time-consuming.

Predicting software defects is an important task in the software life cycle. In addition, an early prediction of the software defect enhances software changes to various setting and enhances the utilization of available. A defect shows the system's unexpected behavior for certain requirements. During software testing, the unexpected behavior is identified and manifest as a defect. A software defect can be called "a flaw in the process of software development that would result in software failing to congregate the preferred anticipation"[2]. In addition, finding and correcting defects leads to costly software development activities [3]. A small number of modules have been observed to contain the mainstreams of software defects [4],[5].

Thus, sensible software defect classification facilitates the efficient allocation of testing resources and facilitates developers to improve a system's architectural design by identifying the system's high - risk segments [6],[7],[8].

 DOI : 10.18231/2454-9150.2019.0256

Machine learning techniques can be used to analyze data from different perspectives and to retrieve useful information from developers. Classification and clustering can be the machine learning techniques that can be used to identify defects in software datasets. It involves categorizing software modules into defective or non-defective software modules that are denoted by a set of software complexity metrics using a classification model resulting from data from earlier development projects [9]. The software complexity metrics may consist of code size [10], the cyclomatic complexity of McCabe [11] and Halstead's Complexity [12].

Finally, the main contribution of the paper is to device a new Relevance based Fuzzy C Means (RFCM) algorithm which efficiently classifies and predicts the accuracy of software defect detection. This algorithm also makes use a heuristic feature selection using fitness function. In Section 2, we present basic preliminaries and related literature work associated with software defect detection. In section 3 the step wise description of the proposed approach is elicited. Dataset description, evaluation methodology and Comprehensive analysis is presented in section 4 and conclusion and future work will be in section 5.

II. BASIC PRELIMINARIES AND RELATED RESEARCH WORK

Most of the existing software defect prediction studies in the literature are limited in carrying out relative empirical analysis of all learning methods. Some of them have used few methods and provide an association among them and others just discussed or proposed a method by extending them based on accessible learning techniques [17].

There are many studies of prediction of software defects using techniques of machine learning. For example, a linear Regression approach was proposed in the study in [2] to predict the faulty modules. With the available historical data of the software accumulated defects, the research study predicts future software faults. Also evaluating and comparing the regression model and POWM model uses a Root Mean Square Error measure.

A new framework for predicting software defects on different datasets over existing classification algorithms is proposed in [13] and pragmatic that their selected classification methods present good predictive accuracy and support the classification based on metrics. For comparison, the operating characteristics of the receiver curve (AUC) are used [14], [15].

Particularly for proportional study in the detection of software defects, it is suggested to use AUC as the primary display of accuracy because it separates extrapolative performance from cost distributions and class, and they are specific characteristics of the project that may be subject to change and unknown. In particular, previous findings concerning efficacy [16] have been confirmed from defect prediction. The results of the experiments showed that in the performance of different classification algorithms there is no significant difference. The study covered only the model of classification for software defect prediction.

Fault prediction of different Machine Learning method are studied in [3], [4] for analyzed the applicability.

A systematic review is presented in [5] using Machine Learning for software defect prediction techniques It shows a comprehensive analysis of all machine learning algorithms and statistical techniques and their study in predicting in software defects and their performances, evaluations compared to different machine learning algorithms with summarized strengths and weakness.

The paper provided a benchmark in [6] to enable a common and useful comparison of different approaches to defect prediction. Developed a defect prediction system (SBPS) model for object-oriented software [7] for defect prediction. The study used the performance measure such as accuracy to evaluate the proposed model. Finally, the results of the study showed that the average model accuracy proposed is 76.27 %.

This present research will discusses well-known machine learning techniques Naïve Bayes, Random Forest, Radial Basis Function, K-nearest Neighbor, Support Vector Machine, K-means clustering Algorithm, Fuzzy c-Means clustering algorithm. It also elucidates the evaluation of machine learning algorithms using a variety of performance measures such as Accuracy, F-measure and Root Mean Square Error (RMSE).

III. RFCM- RELEVANCE BASED FUZZY C-MEANS ALGORITHM

The study aims to examine and review the accuracy of the software defect detection using computational intelligence algorithm known as Relevance based fuzzy c-means algorithm. The present research will shows the performance and ability of the RFCM algorithm in software defect detection and presents an empirical examination of the with the existing state of the art literature algorithms. Following are summarized description of the proposed RFCM algorithm.

A. Feature Selection using Heuristic Fitness Function

Feature selection is an important task in software defect prediction. Because of the diverse nature in the database and containing different types of attributes will introduce abundant attributes in the dataset and the mining task will become composite. In this paper, a Heuristics based Fitness Function (HFF) is used to assess the eminence of each feature [9]. Consider each feature in the dataset with a

 DOI : 10.18231/2454-9150.2019.0256

sequence of data items; apply Fitness function to calculate eminence of each feature using the fitness.

$$Fitness = 1 - \left[\frac{\sum_{i=1}^{n}\left(\frac{SUMi}{C}\right)^2}{N}\right]$$

Where N is the number of features in the dataset, SUMi is the sums of all random sequences in bin i, and bin capacity is C. The features whose Fitness value is less than the threshold limit will be removed from the dataset.

B. Relevance based Fuzzy C-Means algorithm

Relevance based is a feature of certain information retrieval systems. The connotation of based is to obtain the user vector values and use the information to analyze whether the results are relevant to a new query. Relevance based supports the general quadratic distance metrics. It facilitates interactive learning for refining the results. The objects similarity may be described as the degree of relevance among pair wise objects similarity along with the highest rank allied with dimensions of dataset.

One among the adopted solutions was to directly derive the benefit of the human proficiency to access the retrieval results which is based on relevance. The user, for certain retrieval objects, has to offer feedback by spotting the results as relevant or non- relevant data objects. With this information, the system then repeatedly computes an enhanced depiction of the desired information and also retrieval is further refined.

The relevance based approach of ranking construes the hypothesis of uniform distribution of the tuples, however, involves equal significance to every tuple through the class label distribution. The comprehensive performance of the employed technique is stimulated by the score of input data set. In consideration of the optimistic and pessimistic tuples with their allied vital factors the object is updated by adjusting the position of original object in the n-dimension space. Feature Relevance Estimation (FRE) approaches is another example where in a given query; the user may deem some precise objects more vital than others. A significant weight is specified to every object so that object with improved variance has lesser significance compared to objects with lesser variations.

The FCM is the largely a dominant and reputed method in the application of cluster analysis. In the fuzzy clustering, data objects can be assigned to more than one cluster besides linking to each object with a set of membership degrees. The degree of association strength between the data object and a particular cluster is indicated. Fuzzy clustering is a better option in respect of real world cases where no sharp edges exist between clusters. The finite set of data of Crisp requisite partition entails replacement by a pathetic requirement of a fuzzy partition. However, in fuzzy clustering, the fuzzy pseudo partition is a difficult on the same set. Through this technique, the objective function value is optimized and subsequently estimates a known fuzzy assignment of objects to clusters. The data can be separated through this technique , as a fixed group of objects $x_1,x_2,x_3,\ldots,x_n = X$ considering the specified criterion to cluster fuzzy parameter.

C. Notations

$U= [\mu_{ik}]_{cxn}$: Matrix representing classification.

μ_{ik} : data object having k as degree of membership to cluster i.

J(p) : Object function value

m : fuzzy parameter=2

V_i : i being the center of cluster.

x_k : as an object in a data vector.

$\|xk - vi\|^2$: Euclidean distance

The resultant clusters are bestowed upon matrix $[\mu_{ik}]_{cxn}$ where μ_{ik} is the data object with k as degree of membership to cluster i . The corresponding fulfilling conditions are:

$0 \le \mu_{ik} \le 1$ i=1, 2… c k=1, 2… n

$\sum_{k=1}^{c} \mu_{ik} = 1$ k=1, 2… n

$0 < \sum_{k=1}^{n} \mu_{ik} \le n$ i=1, 2… n

Cluster center:

$$V_i = \frac{\sum_{k=1}^{n}[\mu_{ij}]^m x_k}{\sum_{k=1}^{n}[\mu_{ij}]^m}$$

i=1, 2… c

Membership degree:

$$\mu_{ik}(t+1) = \left[\sum_{j=1}^{c}\left[\|x_k - V_i^{(t)}\|^2 / \|x_k - V_i^{(t)}\|^2\right]^{1/m-1}\right]^{-1}$$

Where, μ_{ik} is data object with k being the degree of membership to cluster i. This technique assumes, the number of desired clusters as c, the stopping criterion as a, real number as m along with the respective distance.

D. Pseudo-Code of RFCM Technique

RFCM pseudo-code can be represented as:

- Initially, the constant k is specified by the user in the clustering phase and clustering of an unmarked data point through calculation of the distance between the query point and all the points in the dataset.

 DOI : 10.18231/2454-9150.2019.0256

- The selection of initial fuzzy partition as p(0) and iteration as t=0..
- The evaluation center of cluster c through velocity v1(t),v2(t),…,vc(t) applying the cluster center equation of partition p(t) and the value chosen as m.
- Distances are calculated between the cluster seed point and points in the user vector.
- Estimate μi(t+1) using membership function to validate p(t+1).
- Measure p(t+1) to p(t) and stop if p (t +1) − p (t) ≤ Є otherwise, increase t by one and return to second step .
- If the neighbor lies in a particular class, then it is considered as true positive otherwise it is taken as true negative. Accuracy is calculated based on the true positive and true negative value.

This pseudo-code selects the fuzzy parameter m greater than one, is considered for any problem. The choice of partitioning is difficult with increase in m and there is no evidence to select this value.

IV. EXPERIMENTAL ANALYSIS

In this section, the performance of the RFCM approach is evaluated based on the commonly used accuracy and the elapsed time is measured as the time duration. The datasets are used for evaluation in this is study are five datasets, namely DS1, DS2, DS3, ECLIPSE and MOZILLA. The proposed RFCM clustering technique marks the data with class labels into seven different classes; Blocker, Critical, Enhancement, Major, Minor, Normal, and Trivial.

In order to evaluate the performance of using Machine Learning algorithms Naïve Bayes, Random Forest, Radial Basis Function, K-nearest Neighbor, Support Vector Machine, K-means clustering Algorithm, Fuzzy c-Means clustering algorithm. The paper also evaluates the ML classifiers using various performance measurements i.e. accuracy, F-measure and Root Mean Square Error [16] based on the generated confusion matrixes. The following are measures used for evaluation.

- Accuracy
- Precision (Positive Predictive Value)
- Recall (True Positive Rate or Sensitivity)
- F-measure
- Root-Mean-Square Error (RMSE)

The accuracy of RFCM algorithm is evaluated with existing classifiers on different datasets is shown in Table 1. The RFCM algorithm achieved a elevated accuracy rate. The average value for the accuracy rate in all datasets for the classifiers is over 95% on average. However, the lowly value emerged for KNN algorithm. We believe this is because the dataset is small and RFCM algorithm needs a bigger dataset in order to achieve a higher accuracy value.

Table 1: Accuracy measure for the different algorithms over different data sets

Dataset	Bayes	RF	RBF	KNN	SVM	K-MEANS	FCM	RFCM
DATASET1	84.45	91.56	90.33	65.92	91.97	90.02	93.8	95.11
DATASET2	85.25	86.39	86.38	75.13	86.01	83.65	95.4	97.2
DATASET3	85.9	89.9	89.7	84.24	90.52	86.58	96.3	98.45
ECLIPSE	84.78	82.56	84.68	79.03	82.3	80.99	91.22	93.22
MOZILLA	86.17	89.65	90.87	60.59	90.8	87.91	89.94	90.11

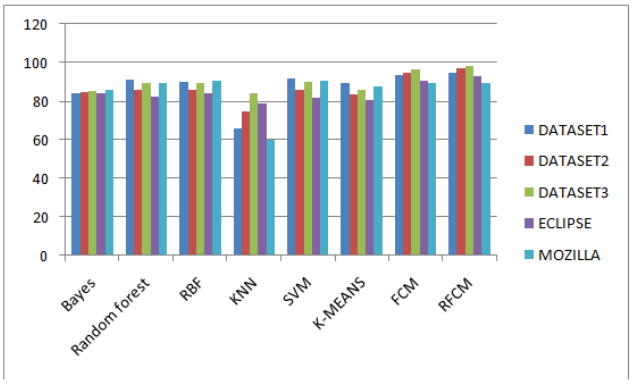

Fig 1: Accuracy measure for the different algorithms over different data sets.

The Fig 1 shows the accuracy of RFCM algorithm with existing classifiers on different datasets. From Fig 1, it is observed that RFCM has attained accuracy measure compared to other classifiers. The average value for the accuracy rate in all datasets for the three classifiers is over 95% on average. However, the lowest value appears for KNN algorithm. We believe this is because the dataset is small and RFCM algorithm needs a bigger dataset in order to achieve a higher accuracy value.

Table 2: f-measure for the different algorithms over different data sets

Dataset	Bayes	RF	RBF	KNN	SVM	K-MEANS	FCM	RFCM
DATASET1	98.4	100	94.9	79	96	94	100	100
DATASET2	96	92	92	84	93	90	98	99
DATASET3	91	94	95	91	95	93	99	100
ECLIPSE	90	89	90	86	90	88	89	92
MOZILLA	91	94	95	72	95	93	94	96

The F-Measure of RFCM algorithm is evaluated with existing classifiers on different datasets is shown in Table 2. From Table 2, the RFCM algorithm achieved a 100% high accuracy rate in Dataset1 and Dataset3. It has attained 94% accuracy in ECLIPSE and MOZILLA dataset.

 DOI : 10.18231/2454-9150.2019.0256

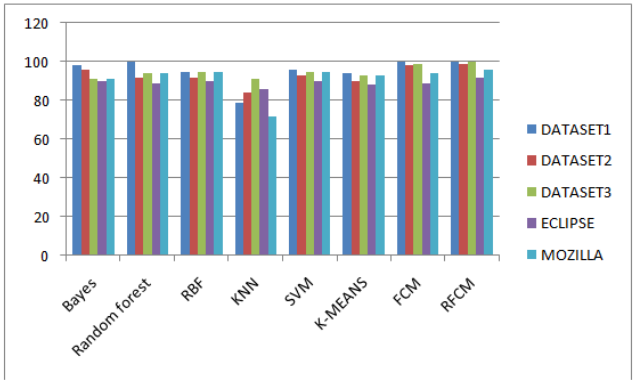

Fig 2: F-measure for the different algorithms over different data sets

In classify to evaluate the existing classifiers through respect to the F-measure value. Fig. 2 shows the F-measure values for the used Machine Learning algorithms in the five datasets. As shown the figure, RFCM has the highest F-measure value in all datasets followed by Random Forest, then Naïve Bayes classifiers.

Table 3: Root Mean Square Error for the different algorithms over different data sets

Dataset	Bayes	RF	RBF	KNN	SVM	K-MEANS	FCM	RFCM
DATASET1	0.17	0.18	0.22	0.32	0.14	0.17	0.15	0.08
DATASET2	0.19	0.28	0.32	0.25	0.18	0.18	0.13	0.1
DATASET3	0.18	0.26	0.27	0.16	0.15	0.19	0.16	0.06
ECLIPSE	0.16	0.22	0.23	0.21	0.19	0.21	0.11	0.09
MOZILLA	0.15	0.14	0.25	0.39	0.09	0.12	0.09	0.07

Table 3 represents Root Mean Square Error for the different algorithms over different data sets. Finally, to assess the Machine Learning algorithms with other approaches, we considered the RMSE value. RFCM representation to forecast the accumulated with software defects. It evaluated their approach with the machine learning model based on the RMSE measure. The evaluation process was done on the different datasets. Fig 3 shows Root Mean Square Error for the different algorithms over different data sets. From figure it is noticed that RFCM has lowest RMSE with respect to existing machine learning algorithms.

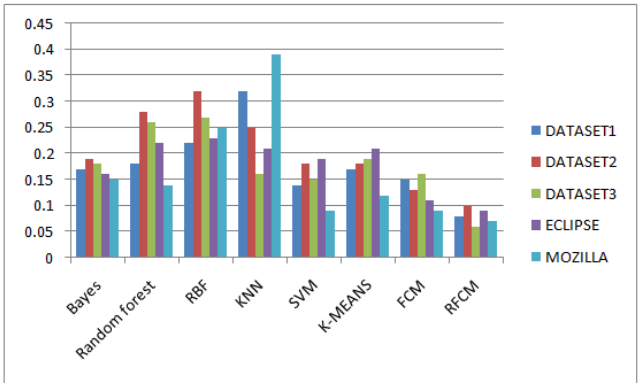

Fig 3: Root Mean Square Error for the different algorithms over different data sets

V. CONCLUSION

Software defect prediction is a technique in which a model is created to predict software defects based on historical data. Different approaches were suggested by means of various datasets with different metrics and performance measures. This paper proposed a adaptive computation intelligence approach known as Relevance based Fuzzy C-Means clustering algorithm and evaluated software defect prediction model and its performance with the using of machine learning algorithms Naïve Bayes, Random Forest, Radial Basis Function, K-nearest Neighbor, Support Vector Machine, K-means clustering Algorithm, Fuzzy c-Means clustering algorithm. Using five real defect datasets, the evaluation process is implemented.

Experimental results are gathered based on measurements of accuracy, F-measure, and Root Mean Square Error (RMSE). The Experimental setup reveals that the RFCM technique is giving highest accuracy and f-measure with respect to all the listed algorithms. The Proposed approach is giving lowest RMSE with respect to the other approaches presented in the paper. The RFCM is an effective approach for predicting software defects. The comparison outcome showed that the RFCM classifier has attain the best consequences over the others. We can involve other Machine Learning techniques as a future work and provide an extensive comparison among them. Furthermore, adding more software metrics in the learning process is one possible approach to increase the accuracy of the prediction model.

ACKNOWLEDGMENT

The first author is thankful to Rayalaseema University, Kurnool and CMR Technical Campus, Hyderabad for providing extensive support for carrying this research work.

REFERENCES

[1] Y. Tohman, K. Tokunaga, S. Nagase, and M. Y., "Structural approach to the estimation of the number of residual software faults based on the hyper-geometric districution model," IEEE Trans. on Software Engineering, pp. 345–355, 1989.

[2] S. Kumaresh & R. Baskaran (2010) "Defect analysis and prevention for software process quality improvement", International Journal of Computer Applications, Vol. 8, Issue 7, pp. 42-47.

[3] K. Ahmad & N. Varshney (2012) "On minimizing software defects during new product development using enhanced preventive approach", International Journal of Soft Computing and Engineering, Vol. 2, Issue 5, pp. 9-12.

[4] C. Andersson (2007) "A replicated empirical study of a selection method for software reliability growth models",

 DOI : 10.18231/2454-9150.2019.0256

Empirical Software Engineering, Vol.12, Issue 2, pp. 161-182.

[5] N. E. Fenton & N. Ohlsson (2000) "Quantitative analysis of faults and failures in a complex software system", IEEE Transactions on Software Engineering, Vol. 26, Issue 8, pp. 797-814.

[6] T. M. Khoshgoftaar & N. Seliya (2004) "Comparative assessment of software quality classification techniques: An empirical case study", Empirical Software Engineering, Vol. 9, Issue 3, pp. 229-257.

[7] T. M. Khoshgoftaar, N. Seliya & N. Sundaresh (2006) "An empirical study of predicting software faults with case-based reasoning", Software Quality Journal, Vol. 14, No. 2, pp. 85-111.

[8] T. Menzies, J. Greenwald & A. Frank (2007) "Data mining static code attributes to learn defect predictors", IEEE Transaction Software Engineering., Vol. 33, Issue 1, pp. 2-13.

[9] Prasanna,K., M.Seetha, and A.P.Siva Kumar, " CApriori: Conviction based Apriori algorithm for discovering frequent determinant patterns from high dimensional datasets", ICSEMR 2014.

[10] D. Shiwei (2009) "Defect prevention and detection of DSP-Software", World Academy of Science, Engineering and Technology, Vol. 3, Issue 10, pp. 406-409.

[11] P. Trivedi & S. Pachori (2010) "Modelling and analyzing of software defect prevention using ODC", International Journal of Advanced Computer Science and Applications, Vol. 1, No. 3, pp. 75- 77.

[12] Awni Hammouri, Mustafa Hammad, Mohammad Alnabhan, Fatima Alsarayrah," Software Bug Prediction using Machine Learning Approach,", in the journal of ijacsa, vol 9 no 2, 2018 pp 78-83.

[13] S. Lessmann, B. Baesens, C. Mues & S. Pietsch (2008) "Benchmarking classification models for software defect prediction: A proposed framework and novel finding", IEEE Transaction on Software Engineering, Vol. 34, Issue 4, pp. 485-496.

[14] K. El-Emam, S. Benlarbi, N. Goel, & S.N. Rai (2001) "Comparing Case- Based Reasoning Classifiers for Predicting High-Risk Software Components", Journal of Systems and Software, Vol. 55, No. 3, pp. 301-320.

[15] L.F. Capretz & P.A. Lee, (1992) "Reusability and life cycle issues within an object-oriented design methodology", in book: Technology of Object-Oriented Languages and Systems, pp. 139-150, Prentice-Hall.

[16] Olsen, David L. and Delen, " Advanced Data Mining Techniques ", Springer, 1st edition, page 138, ISBN 3-540-76016-1, Feb 2008.

[17] S. Adiu & N. Geethanjali (2013) "Classification of defects in software using decision tree algorithm", International Journal of Engineering Science and Technology (IJEST), Vol. 5, Issue 6, pp. 1332-1340. 12.

 DOI : 10.18231/2454-9150.2019.0256