ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

Object Oriented Approach for Analysis of Software Fault Prediction using K-Jensen Shannon Entropy Model based Clustering Algorithm

M.Praneesh[1], K.Mahalakshmi[2]

Assistant Professor, Department of Computer Science, Sankara College of Science and Commerce, India [1]

Assistant Professor, Department of Computer Science, Sankara College of Science and Commerce, India [2]

Abstract: In software engineering, the most frequent problem highlighted by IT Practioners concerned the measurement of quality. In order to improve the quality of the software, fault prediction is the necessary task. This prediction reduces the time complexity between modules. In the recent years lot of software metrics are used for predicting whether the particular models of the software faulty are fault free. In this paper we have proposed K-Jensen Shannon Entropy Model based Clustering Algorithm for predicting the faults in software projects. In our experiment, we used CM1, PC1, KC1, KC2 and PC4 collected from NASA MDP. Finally, our proposed system is compared with Euclidean distance based K-Means Clustering Algorithm.

Keywords: software fault prediction, clustering, Quality and Metrics

I. INTRODUCTION

Software Fault Prediction is an important analysis in software development life cycles, which avoid many problems in software process and improve the quality of the required software and also reduce the time complexity. The main objectives of this prediction whether the software development process, the required models of the software is fault are fault free. Many researchers have already predict massive metrics and techniques like correlation, data mining algorithms, decision tree, neural networks, genetic algorithm, SVM Classification, Naïve Bayes Classification have been analyzed for fault prediction.

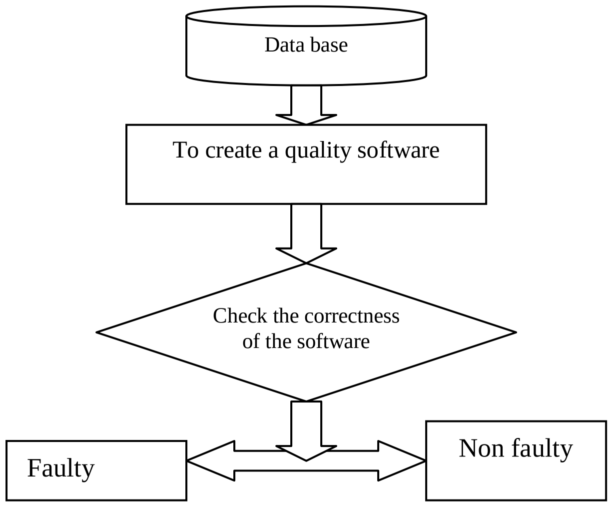

Fig-1 Software Fault Prediction Model

II. RELATED WORKS

Vikas Gupta et al summarized the basic concepts of clustering and analyze the fault prediction based on JEdit open source software. They implemented K-Means clustering based classification of software modules into faulty or non-faulty.

Meenatsh P.C et al proposed fault prediction using EM based Quad tree also not fit to be prediction of software. Ajeet kumar pandey et al predicted faults in various software software model based on Fuzzy logic. The ultimate goal of this paper is to improving the software reliability and portability. Pradeep Singh et al proposed software fault prediction model using clustering based classification based on learning systems.

Turhan et al analyzed the software faults based on Weighted Naïve Bayes classification algorithm which have performed Static code attributes such as lines of code, size of the complexity.Menzies et al proposed Naïve bayes algorithm based on LogNums filter were implemented to achieve the desired results with 71%. Shanthini et al focused high performance analysis based on machine learning approach.

Akalya devi et al analyzed a hybrid feature selection method (Correlation based feature selection, Chi Squared , OneR, and Gain ratio, Naïve Bayes, RBF Network, J48) to be performed. the performance measures like Mean Absolute Error(MAE), Root Mean Squared Error(RMSE).

Hassan Najadat proposed that modified Ripple DOwn Rule learns the defect prediction based on two different algorithm such as CLIPPER and RIDOR. This paper is carried out with static code attributes finally improve the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

quality of the software and high portability with effectiveness.

III. METHODOLOGY

The following steps are the functionality of the paper represented as follow as

- Data set are collected from NASA MDP.
- Construct the Distance measures based clustering algorithm.
- Performance measures are discussed based on proposed algorithm and existing algorithm.
- Comparative analysis are performed based on Euclidean based K-Means Clustering with K-Jensen Shannon Entropy Model based Clustering Algorithm

A. Data set used

Software fault prediction is important feature in improving the software Quality metrics. In this paper we are used CM1, PC1, KC1, KC2 and PC4 collected from NASA MDP. These datasets are publicly available PROMISE project storage. Analyze of this paper we are used five different data. Each data is consisting of number of modules. The quality of this module is described by its error rate. Error rate is described as "number of defects in the module, and Defect, whether or not the module has any defects. The table 1 shows that description about the data set.

TABLE 1: DATA SET

Data	Modules/ instances	Language	Description
CM1	498	C	Space craft instrument
PC1	1109	C	Earth orbiting satellite
KC1	2109	C++	Storage management for ground data
KC2	522	C++	Science data processing
PC4	1458	C	Flight software for earth orbiting satellite

B. Euclidean based K-Means Clustering Algorithm

Let X = {X1, X_2…X_k} be set of data and
M = {m1, m_2…..m_k}

1. Select a number (K) of cluster centers – centroid at random
2. Assign every item to its nearest cluster center using Euclidean distance

$$s_i^t = \{x_j : \|x_j - m_i^{(t)}\| \le \|x_j - m_i^t\|$$

For all i=1…..k}

3. Move each cluster center to mean of its assigned items

$$m_i^{t+1} = \frac{1}{|s_i^{(t)}|} \sum_{x_j \in s_i^{(t)}} x_j$$

4. Repeat steps 2, 3 until convergence yet

Fig-2 K-Means Clustering Algorithm with Euclidean Distance

Distance measure is essential steps in clustering that will verify how the similarity of two elements is calculated. In this paper, K-Means clustering algorithm is based on Euclidean distance measure. In last two hundred decades, Euclid stated that shortest distance between two points.

C. K-Jensen Shannon Entropy Model based Clustering Algorithm

Let X = {X1, X_2…X_k} be set of data and
M = {m1, m_2…..m_k}

5. Select a number (K) of cluster centers – centroid at random
6. Assign every item to its nearest cluster center using Jensen- Shannon distance

$$d_{JS} = \frac{1}{2}\left[\sum_{i=1}^{d} p_i In\left(\frac{2p_i}{p_i + q_i}\right) + \sum_{i=1}^{d} Q_i In\left(\frac{2Q_i}{p_i + Q_i}\right)\right]$$

7. Move each cluster center to mean of its assigned items

$$m_i^{t+1} = \frac{1}{|s_i^{(t)}|} \sum_{x_j \in s_i^{(t)}} x_j$$

8. Repeat steps 2, 3 until convergence yet

Fig-3 K-Jensen Shannon Entropy Model based Clustering Algorithm

D. Performance Measures

The proposed system is evaluated several performance metrics such as true positive rate, false positive rate, precision, recall, F-Measure and accuracy.

True positive rate: This measure is projected by the modules that are predicted positively as the results specified at the end. The general for mat is represented below equation.

True positive rate = true positive rate / (true positive rate + false negative rate

False Positive rate: This measure is projected by the modules that are predicted incorrectly categorized ad class x/ actual total of all classes, except x.

False positive rate = false positive rate / (true negative + true negative rate

Precision: precision gives positive predicate values and it process values or product quality or exactness.

Precision = True positive / (True Positive + False positive)

Recall: recall gives sensitive of problem and it process values or product quantity or completeness. This measure is used to recognize total number of modules.

Recall = true positive / (true positive + false negative)

F-Measure: it is one of the quality measures of the modules. The general formula is represented as given below

F-Measure = 2* Precision * recall / (precision + recall)

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

Accuracy: it is calculated as a number of instances predicted positively divided by total number of instances

Accuracy = (true positive + true negative) / (P+N)

IV. EXPERIMENTAL RESULTS

In our experimental analysis, five different metrics are analyzed namely CM1, PC1, KC1, KC2 and PC4. These dataset contains both structure and object oriented. In this work carried out several performance metrics such as true positive rate, false positive rate, precision, recall, F-Measure and accuracy for evaluated our proposed work with existing system. In our proposed system is effective and high robustness when compared to the Existing methods

TABLE 1: CLASSIFIED INSTANCES FOR CM1

Method	Approximately Classified Instances	Inaccurately classified instances	Total instances
K-Means	425	73	498
Proposed	445	53	498

TABLE 2: PERFORMANCE ANALYSIS FOR CM1

Method/ Performance measures	K-Means	Proposed
TP Rate	0.85	0.90
FP Rate	0.61	0.70
Precision	0.83	0.85
Recall	0.85	0.87
F-Measure	0.85	0.86
Accuracy	0.83	0.88

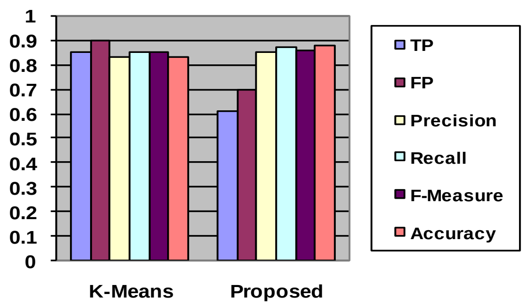

Fig- 4 Performance Analysis for CM1 Data set

TABLE 3: CLASSIFIED INSTANCES FOR PC1

Method	Approximately Classified Instances	Inaccurately Classified instances	Total instances
K-Means	965	144	1109
Proposed	1025	84	1109

TABLE 4: PERFORMANCE ANALYSIS FOR PC1

Method/ Performance measures	K-Means	Proposed
TP Rate	0.92	0.96
FP Rate	0.65	0.69
Precision	0.88	0.90
Recall	0.89	0.90
F-Measure	0.89	0.91
Accuracy	0.89	0.92

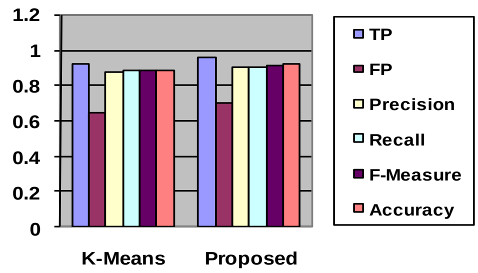

Fig- 5 Performance Analysis for PC1 Data set

TABLE 5: CLASSIFIED INSTANCES FOR KC1

Method	Approximately Classified Instances	Inaccurately classified instances	Total instances
K-Means	965	144	1109
Proposed	1025	84	1109

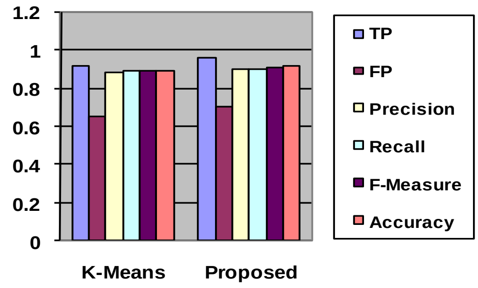

Fig- 6 Performance Analysis for KC1 Data set

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

TABLE 6: PERFORMANCE ANALYSIS FOR KC1

Method/ Performance measures	K-Means	Proposed
TP Rate	0.92	0.96
FP Rate	0.65	0.69
Precision	0.88	0.90
Recall	0.89	0.90
F-Measure	0.89	0.91
Accuracy	0.89	0.92

TABLE 7: CLASSIFIED INSTANCES FOR KC2

Method	Approximately Classified Instances	Inaccurately Classified instances	Total instances
K-Means	470	52	522
Proposed	501	21	522

TABLE 8: PERFORMANCE ANALYSIS FOR KC2

Method/ Performance measures	K-Means	Proposed
TP Rate	0.90	0.94
FP Rate	0.63	0.67
Precision	0.80	0.90
Recall	0.87	0.90
F-Measure	0.88	0.91
Accuracy	0.86	0.94

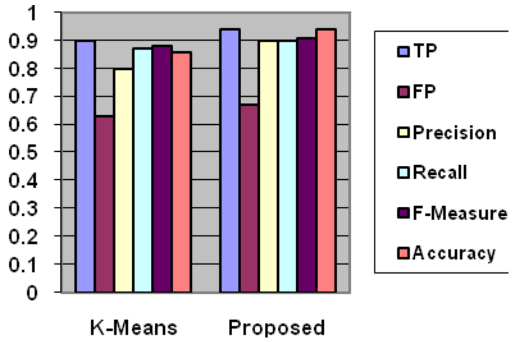

Fig- 7 Performance Analysis for KC2 Data set

TABLE 9: CLASSIFIED INSTANCES FOR PC4

Method	Approximately Classified Instances	Inaccurately Classified instances	Total instances
K-Means	1280	178	1458
Proposed	1325	155	1458

TABLE 10: PERFORMANCE ANALYSIS FOR PC4

Method/ Performance measures	K-Means	Proposed
TP Rate	0.91	0.95
FP Rate	0.70	0.75
Precision	0.89	0.91
Recall	0.87	0.93
F-Measure	0.85	0.95
Accuracy	0.90	0.98

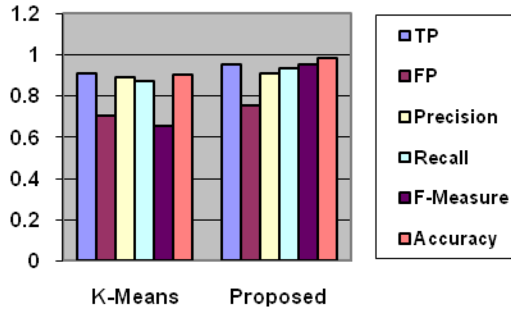

Fig- 8 Performance Analysis for PC4 Data set

V. CONCLUSION

The main intention of this work is to analyze the performance of K-Means clustering algorithm with Euclidean Distance and K-Jensen Shannon Entropy Model based Clustering Algorithm using different metrics of NASA datasets. Based on this performance analysis we conclude that our proposed approach is suitable for small and large data set. The complexity factor is low when compared to the existing approach. The future enhancement of this work is planned to measure different similarity measures with Fuzzy logic approach based on Equivalence and Composite relations.

REFERENCES

[1] A Kaur, et. al. (2009),‖Early software fault prediction using real time defect data‖, 2009 Second International Conference on Machine Vision, pp 243-245

[2] Rashid, Ekbal, Patnayak S, Bhattacherjee V. Estimation and evaluation of change in software quality at a particular stage of software development. Indian Journal of Science and Technology. 2013; 6(10):5370-9.

[3] Sathyaraj R, Prabu S. A survey-quality based object oriented software fault prediction. International Journal of Engineering and Technology. 2013 Jun–Jul; 5(3): 2349-51.

[4] Catal C, Diri B. Investigating the effect of dataset size, metrics sets and feature selection techniques on software fault prediction problem. Information Sciences. 2009; 170(8):1040-58.

[5] Jiang Y, Cukic B, Ma Y. Techniques for evaluating fault prediction models. Empirical Software Eng. 2008; 13(5):561– 95.

[6] Kaur S, Kumar D. Software fault prediction in object oriented software systems using density based clustering approach. International Journal of Research in Engineering and Technology (IJRET). 2012 Mar; 1(2):111-7.

[7] Moeyersoms J, Fortuny EJ, Dejaeger K, Baesens B. Comprehensible software fault and effort prediction: A data

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

mining approach. The Journal of Systems and Software. 2015 Feb; 100:80-90.
[8] Boetticher G. Improving credibility of machine learner models in software engineering. Advanced Machine Learner Applications in Software Engineering. Hershey, PA, USA: Idea Group Publishing; 2006.
[9] NASA Metrics Data Program. 2015 Apr 15. Availablefrom:http://promise.site.uottawa.ca/SERepository/datasets-page.html
[10] Catal C, Sevim U, Diri D. Practical development of an Eclipse-based software fault prediction tool using Naive Bayes algorithm. Expert Systems with Applications. 2011; 38(3):2347-53.
[11] Archana Singh et. al. International Journal of Computer Applications (0975 – 8887) Volume 67– No.10, April 2013
[12] Aditi Sanyal, Balraj Singh, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1, January 2014 ,ISSN: 2277 128X
[13] Teknomo, Kardi, Similarity Measurement Available fromhttp:\\people.revoledu.com\kardi\tutorial\Similarity\
[14] Bray J. R., Curtis J. T., 1957. An ordination of the upland forest of the southern Winsconsin. Ecological Monographies, 27, 325-349.
[15] G.Gan,C. Ma,J.Wu,―Data clustering: theory,algorithms, and applications‖, Society for Industrial and Applied Mathematics, Philadelphia, 2007.
[16] Jiang Y. et. al., ―Fault Prediction Using Early Lifecycle Data‖. ISSRE 2007, the 18th IEEE Symposium on Software Reliability Engineering, IEEE Computer Society, Sweden, pp. 237-246.
[17] Seliya N., Khoshgoftaar T.M. (2007), ―Software quality with limited fault-proneness defect data: A semi supervised learning perspective‖, published online pp.327-324.
[18] Jiang Y, Cukic B, Menzies T,‖Cost curve Evaluation of fault prediction models‖, Proceedings of the 2008 19th International Symposium on Software Reliability Engineering, 2008,pg 197-206
[19] Basili, V.R., Calidiera, G., Rombach, H.D.: Goal Question Metric Paradigm.In: Marciniak, J.J. (ed.): Encyclopaedia of Software Engineering, pp. 528-532, Wiley, New York, 1994.
[20] Catal, Cagatay, and Banu Diri. "A systematic review of software fault prediction studies." Expert systems with applications 36.4 (2009): 7346-7354.
[21] Dubelaar, Chris, Amrik Sohal, and Vedrana Savic. "Benefits, impediments and critical success factors in B2C E-business adoption." Technovation25.11 (2005): 1251-1262.
[22] Graves, Todd L., et al. "Predicting fault incidence using software change history." IEEE Transactions on software engineering 26.7 (2000): 653-661. IEEE Standard Classification for Software Anomalies," in IEEE Std 1044-2009 (Revision of IEEE Std 1044-1993) , vol., no., pp.1-23, Jan. 7 2010, doi: 10.1109/IEEESTD. 2010. 5399061.
[23] Jiang, Yue, Bojan Cukic, and Tim Menzies. "Fault prediction using early lifecycle data." The 18th IEEE International Symposium on Software Reliability (ISSRE'07). IEEE, 2007. Sommerville, Ian. "Integrated requirements engineering: A tutorial." IEEE software 22.1 (2005): 16-23.
[24] Todd L. Graves, Alan F. Karr, J.S. Marron, and Harvey Siy, "Predicting Fault Incidence Using Software Change History" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 26(7), 653-661, JULY 2000.
[25] Zelkowitz, M.V., Wallace, D.R.: Experimental models for validating technology. IEEE Computer, (31)5, pp. 23-31, May 1998.
[26] Ma, Y., Guo, L. (2006), "A Statistical Framework for the Prediction of Fault-Proneness", West Virginia University, Morgantown.
[27] Thomas Zimmermann, Nachiappan Nagappan, " Predicting Defects Using Social Network Analysis on Dependency Graphs", International Conference on Software Engineering (ICSE 2008), Leipzig, Germany.
[28] Audris Mockus, Nachiappan Nagappan and Trung T.Dinh-Trong "Test Coverage and Post-Verification Defects: A Multiple Case Study," ACM-IEEE Empirical Software Engineering and Measurement Conference (ESEM), Orlando, FL, 2009
[29] Cagatay Catal & Banu Diri , " A Systematic Review of Software Fault Prediction Studies" Journal of Expert Systems with Applications, Volume 36, Issue 4,May 2009.
[30] Jonas Boberg , " Early Fault Detection with the Model-based Testing" , 7th ACM SIGNPLAN workshop on ERLANG, 2008.
[31] Bindu Goel & Yogesh Singh ,"Emperical Investigation of Metrics for Fault Prediction on Object Oriented Software" the Book series in Computational Intelligence, 2008.
[32] Khoshgoftaar, T. M., Allen, E. B., Ross, F. D., Munikoti, R., Goel, N. & Nandi, A., "Predicting fault-prone modules with case-based reasoning". ISSRE 1997, the Eighth International Symposium on Software Engineering (pp. 27-35), IEEE Computer Society (1997).

BIOGRAPHIES

Praneesh.M received the M.Sc Degree in Computer Science from Bharthiar university, Coimbatore, India in 2010, the Master of Philosophy in Computer from Bharthiar university, Coimbatore, India in 2011, and the Post Graduate Diploma in Natural Language Processing (PGDNLP) and Post Graduate Diploma in Data mining (PGDDM) from Annamalai University, Chennai, India in 2013. He is Presently Assistant Professor and Research Coordinator, Department of Computer Science, Bharthiar University, Sankara College of Science and commerce. His research interests are Image Processing, Remote sensing, Data mining, Natural language processing and Software Metrics/Reliability. He has Published 31 Journals in National / International levels & also he presented and published more than 95 research articles in National and International Conferences. He has published five books: Research Ethics in Computer Science, Discrete Mathematics, and Writing for visual media, Global Trends in Information Systems and Software Applications and fundamentals of Animation. He is a Member of ACM, IAENG, SDIWC, SCIEI, IACSIT, CSTA and ASR.

K.Mahalakshmi received the M.Sc Degree in Computer Science from PSG arts and Science College, Coimbatore, India in 2009, the Master of Philosophy in Computer from Bharthiar University, Coimbatore, India in 2012; she is presently working as Assistant Professor, Department of Computer Science and Applications, Sankara College of Science and commerce. His research interests are Software Engineering, Image and Video Processing. She has Published 4 Journals in National / International levels & also he presented and published more than 12 research articles in National and International Conferences. She is a Member of IAENG, SDIWC, CSTA and IACSIT.