Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue5,May-2017

Software Fault Detection using the Data Pre-Processing and Support Vectore Machine

Hirali Amrutiya[1] , Riddhi Kotak[2], Mittal Joiser[3]

[1] *Research scholar, Computer Engineering MEFGI,*
[2] *Assistance prof., Information technology MEFGI, Rajkot*
[3] *Assistance prof., Computer Engineering MEFGI, Rajkot*

*Abstract-**With increases use of high speed internet huge organization like banking, hospital and industrial work are done using the software rather than manual work. So this organization are required high quality software. If failure happen in the system is effect the financial cost of the organization. Software fault detection model is used for to identify the fault in software which cause the failure. In this paper fault detection model is proposed which used data pre-processing and support vector machine classifier.in data pre-processing feature ranking is perform using information gain and feature cluster is form using fuzzy c-means clustering.***

Keywords: Data mining, data pre-processing, fuzzy c-means, software fault detection, SVM.

I INTRODUCTION

The growing use of the software in daily life is need to provide the high quality software at low cost. Software fault detection is important activity for providing the high quality software. Because of the software fault detection model decrease the effort cost for maintenance of the software. Because of the software fault detection model used limited testing resources at the testing phase of the software development life cycle.

Software fault detection is done by classifying the faulty or non-faulty module. Software metrics are used for identify the fault or non-fault module of the software. Metrics are compute form the source code of the software. This metrics value is called feature of the software. Data mining techniques like Naïve Bayes, Neural Network, Support Vector Machine and genetic programming are used to identify the faulty module. Data pre-processing is increased the result of the supervised learner. Public NASA datasets are used for software defect detection which is collect from the PROMISE repository.

The objective of the paper is to design the software fault detection model which includes data pre-processing and support vector machine. In this paper includes the methodology, proposed framework with results and comparison with another techniques.

II RELATED WORK

The quality of dataset is improved by data pre-processing, which incorporates feature selection and sampling which reduce instances. In feature selection is the method of distinctive and removing irrelevant and duplicate features from a dataset in order that which increase the performance of classification model. [1]-[2]

Wangshu Liu et al. used data pre-processing approach which consists the two stage used for identify the fault in software. Compare result using the Naïve Bayes, C4.5, IB1.[3]

Rohit Mahajan et al. proposed the framework for software fault prediction using the Bayesian Regularization (BR) and compare with Levenberg-Marquardt (LM) and Back propagation (BPA). Bayesian regularization give better performance. [4]

Gupta, Deepika, Vivek K. Goyal, and Harish Mittal Proposed Estimating of Software Quality with Clustering Techniques. This paper focus on clustering with very large dataset and very many attribute of different types. Effective result can be produced by using fuzzy c-mean clustering. [5]

Arashdeep Kaur et al. Proposed model for software fault prediction. In this paper, investigate the fuzzy c mean and k-mean performance. Fuzzy c mean is better than k- mean for requirement and combination metric model. Also investigate the metrics used in early life cycle can be used to predict fault module or not.[6]

Wangshu Liu et al. use the clustering based feature selection method for software fault prediction. FF-relevance and FF- Correlation Measure use. Heuristics approach use for cluster formation. [7]

Issam H. Laradji et al. use the greedy forward feature selection and Average Probability Ensemble learning model is to classify data. This model contain seven algorithm such as W-SVM, Random Forest etc. [8]

III METHODOLOGY

3.1. Data Pre-Processing

In data pre-processing step feature ranking is perform using information gain. The feature ranking rank the feature based on the information that provide. Correlation between the feature and class measure to rank the feature. For measuring the correlation information gain (IG) use. Information gain measure the amount of information provided by the feature f, whether instance is fault or non-fault. The formula used for measuring the information gain is: [3]

$$IG(f) = H(A) - H(A|B)$$

Where $H(A)$ compute the entropy of the district random variable A (i.e. class). Consider p (a) denote prior probability of a value a of A then $H(A)$ compute by formula:

$$H(A) = -\sum_{a} \in_{a} p(a) \log_2 p(a)$$

H (A|B) compute the conditional entropy which quantifies the uncertainty of A given the observed variable B. Consider p (a|b) denote posterior probability of a for value b. H (A|B) compute by formula:

$$H(A|B) = -\sum_{b} \in_{B} P(b) \sum_{a} \in_{A} p(a|b) \log_2 p(a|b)$$

3.2. Fuzzy C-Means Clustering

In second step of the framework cluster will generate using the fuzzy c- means clustering method. Fuzzy c-mean clustering moves the data centre iteratively to the right location.

Fuzzy c- means (FCM) generate the cluster in which allow to data in belongs to more than one cluster. The minimize value of the objective function is used for generate the cluster. Formula for objective function is given below:

$$J_m = \sum_{i=1}^{N} \sum_{J=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \quad ,1 \le m < \infty$$

Where, m = real number greater than 1.
uij = membership degree of xi in the cluster j.
xi = ith at d-dimensional measured data.
cj = d-dimension Center of the cluster.

Iterative optimization of the objective function is carried out in FCM. During the iteration process cluster centers cj and membership uij update by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

This iteration process will stop when $max_{ij} \left\{ |u_{ij}^{(K+1)} - u_{ij}^{(k)}| \right\} < \varepsilon$.

Where, k are steps of iteration and is a termination value between 0 and 1.

3.3. Support Vector Machine

Support vector machine (SVM) used as the classifier for the software defect detection. Support vector machine is supervised learning algorithm. Liner function is used in SVM. So it is work ad binary classification. It is classify the software module in two class Fault and Non- Fault.

IV PROPOSED SOLUTION

Figure 1 shows the proposed framework . this framework have two step first step are data pre-processing in which information gain and fuzzy c-means clustering. In second step classification will perform using support vector machine. Database are taken from promise data repository.

In proposed framework feature are rank using the information gain. After that clustering of the feature are done. For that fuzzy c-means clustering is used. After that cluster are give the input to the classification model. For the implementation of the proporsed framework MATLAB and WEKA 3.6 used.

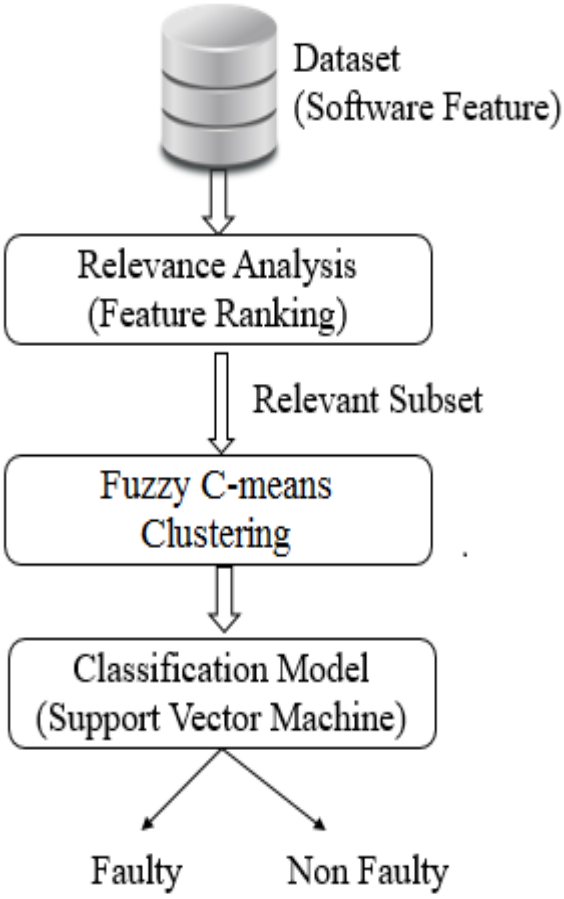

Figure 1. Proposed Framework

V EXPERIMENTAL RESULT AND ANALYSIS

The implementation of data mining algorithms requires the more powerful software tools. The number of available tools for data mining is grow therefor the choice of the suitable tools becomes difficult. We have used WEKA 3.6 and MATLAB for implementing proposed work.

5.1. Dataset

Features are extracted from the source code of the software. For software detection model collect the dataset from the promise data repository. Take CM1 dataset from the repository. Which contain the 21 features of the software and 498 instances. All the feature value is numeric. The basic feature avialble in dataset are:

- Line of code(Loc)
- Haslated complexity metrics
- Mccabe Complexity metrics
- Unique oprands
- Unique opreater
- Total operands
- Total operater

5.2. Analysis of the Result

First step of the proposed framework is to count the information gain and raking the feature. Figure 2 shows the information gain results for the CM1 dataset.

Second step of the proposed framework is to generate cluster of the feature using the Fuzzy C-means algorithms. In this step three cluster will generate which will show in figure 3.

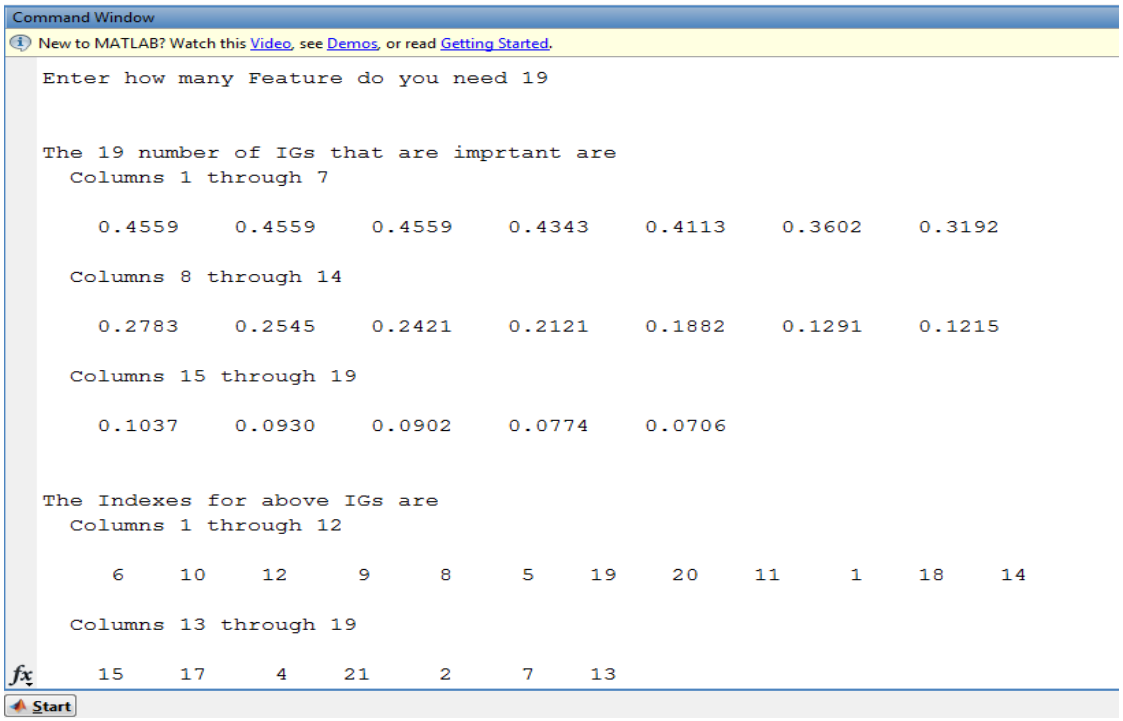

Figure 2 information gain for CM1 dataset

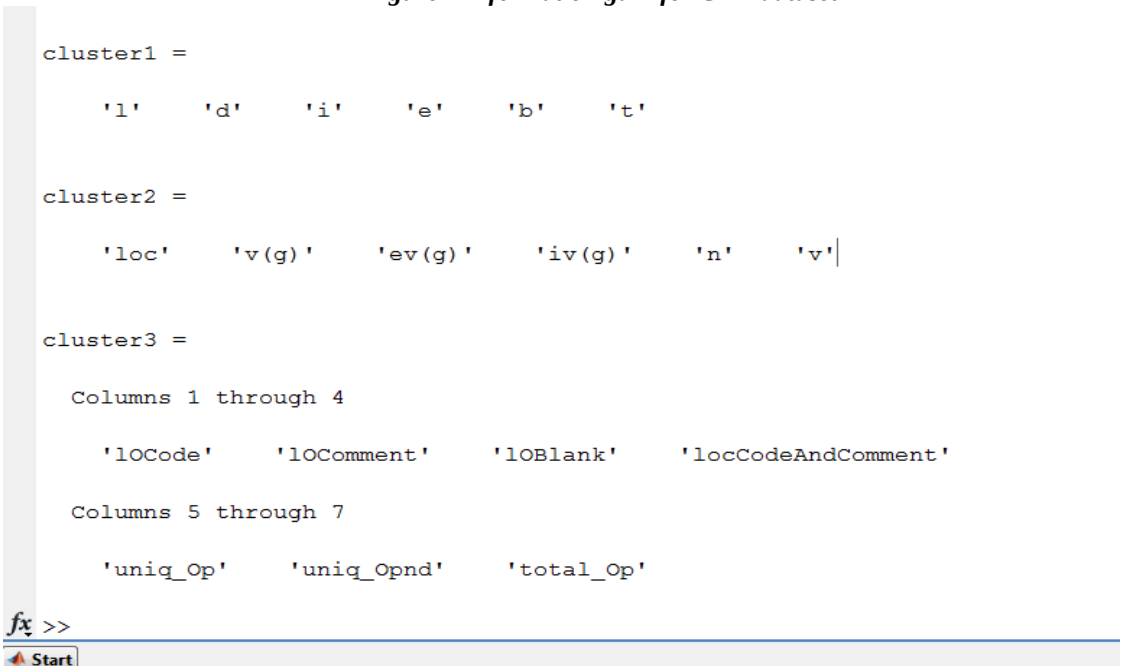

Figure 3 Cluster of features

In the third step classification model is used for the classifying the fault and non-fault modules. For that different classifier is applied directly to the CM1 dataset and compare that result. Here we consider the three classifier which are Neural Networks, Naïve Bayes and Support Vector Machine. Results of this Classifier are show in table 1.

Algorithm	**Accuracy**
Neural Network	68.7
Naïve Bayes	77.7778
Support Vector Machine(SVM)	81.82

Table 1. Accuracy of algorithms

Form this results identify that the support vector machine gives the better accuracy for software fault detection dataset. So we used the SVM as classifier in proposed framework. The results for the proposed framework is shown in table 2 and figure 4.

Performance Measure	Cluster1	Cluster2	Cluster3	Data
Accuracy	86.8687	84.8485	83.8384	81.8182
Precision	0.9425	0.987	0.9205	0.9737
Recall	0.9111	0.8444	0.9	0.8222

Table 2. Results of proposed framework

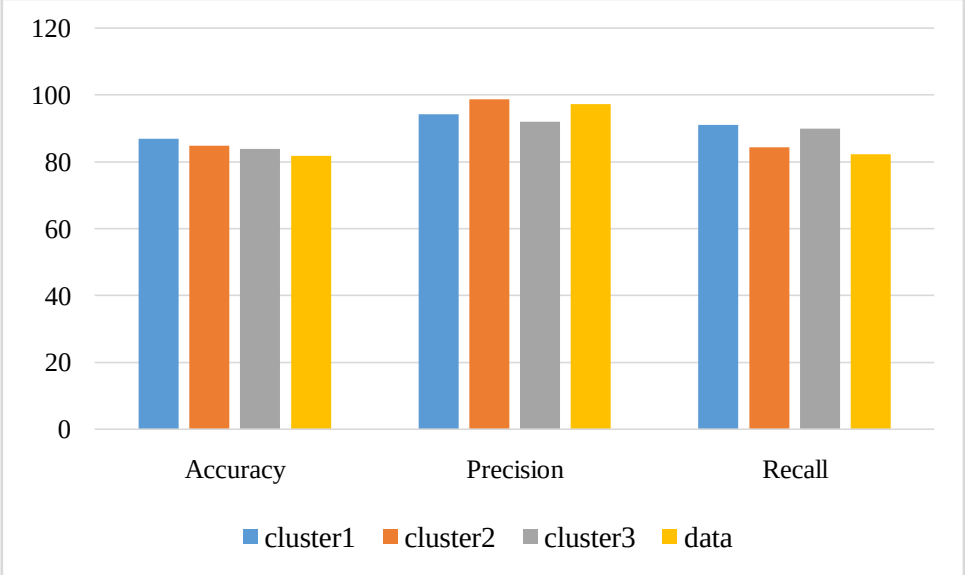

Figure 4. Results of proposed Framework

Figure 5 shows the comparison graph of the proposed framework and existing solution. For that accuracy of the proposed framework is better.

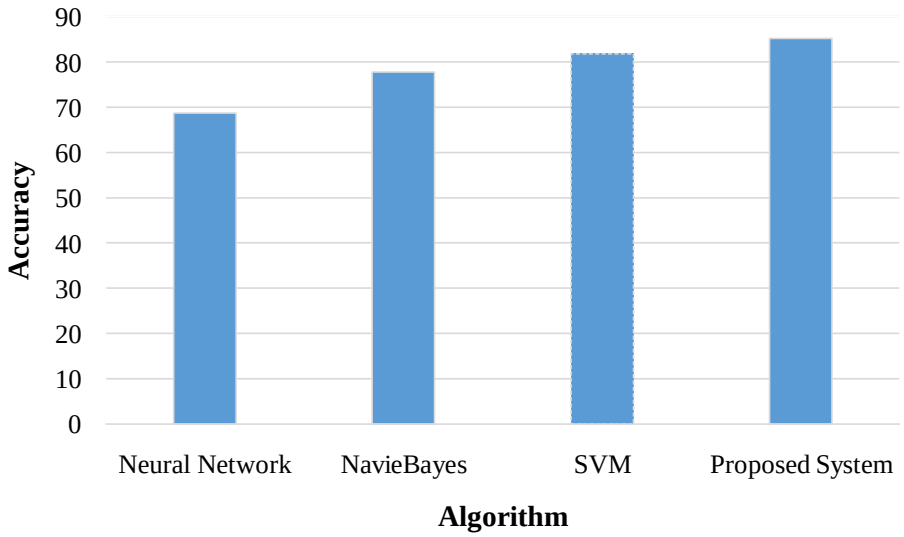

Figure 5. Comparison of Results

VI CONCLUSION

In this research paper we introduce the software fault detection model for identify the faulty modules in software. Also comparing the results of the proposed framework and exiting classifier and from that we can say that proposed framework gives the better results.

REFERENCES

[1] K. Gao, T. M. Khoshgoftaar, H. Wang, and N. Seliya, "Choosing software metrics for defect prediction: An investigation on feature selection techniques," Softw.-Practice Exper., vol. 41, no. 5, pp. 579–606, 2011.
[2] Shivaji, E. J. W. Jr., R. Akella, and S. Kim, "Reducing features to improve code change-based bug prediction," IEEE Trans. Softw. Eng., vol. 39, no. 4, pp. 552–569, 2013.
[3] J. Chen, S. Liu, W. Liu, X. Chen, Q. Gu, and D. Chen," Empirical Studies of a Two-Stage Data Preprocessing Approach for Software Fault Prediction" in IEEE Trans. on Reliability , Vol. 65, pp. 38-53, 2016.
[4] Rohit Mahajan, Sunil Kumar Gupta, Rajeev Kumar Bedi, "Design Of Software Fault Prediction Model Using BR Technique," in Proc. Int. Conf. Information and Communication Technologies, Kochi, pp. 849-858, 2014.
[5] Gupta, Deepika, Vivek K. Goyal, and Harish Mittal. "Estimating of Software Quality with Clustering Techniques," in Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on. IEEE, 2013.
[6] Kaur, Arashdeep, Amanpreet Singh Brar, and Parvinder S. Sandhu. "An empirical approach for software fault prediction," in Industrial and Information Systems (ICIIS), International Conference on. IEEE, pp. 261-265, 2010.
[7] W. Liu, X. Chen, S. Liu, D. Chen, Q. Gu, J. Chen, "FECAR: A Feature Selection Framework for Software Defect Prediction," in proc. Int. Computers, Software & Applications Conference, pp. 426-435, 2014.
[8] Issam H. Laradji, Mohammad Alshayeb, Lahouari Ghouti," Software defect prediction using ensemble learning on selected features,", in Information and Software Technology, Vol. 58,pp. 388-402, 2015.
[9] Ritika sharma, Neha Budhija, Bhupinder singh, "Study of predicting Fault Prone Software Modules," in International Journal of Advanced Research in Computer Science and Software Engineering ,vol. 2, no. 2 pp.1-3, Feb 2012.
[10] "PROMISE SOFTWARE ENGINEERING", http://promise.site.uottawa.ca/SERep ository/datasets-page.html.
[11] A. Soleimani, F. Asdaghi, "An AIS Based Feature Selection Method for Software Fault Predictions" in iranian conferences on intelligent system (ICIS) IEEE, Iran, 2014.
[12] Shanthini. A, Chandrasekaran.RM," Analyzing the Effect of Bagged Ensemble Approach for Software Fault Prediction in Class Level and Package Level Metrics" in int. conf. on information communication and embedded system(ICICES2014) IEEE, Chennai, 2014.
[13] Santosh Singh Rathore, Sandeep kumar, "Comparative Analysis of Neural Network and Genetic Programming For Number of Software Faults Prediction" in IEEE National Conference on Recent Advances in Electronics & Computer Engineering, Roorkee, 2015.

