IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 5, May 2022

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.118

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

Software Defect Estimation Using Machine Learning Algorithms

Shama Parveen[1], Mrs. Sumangala Patil[2]

PG Student, Department of CSE, Lingaraj Appa Engineering College Bidar, Karnataka, India[1]

Associate Professor, Department of CSE, Lingaraj Appa Engineering College Bidar, Karnataka, India[2]

ABSTRACT: Software metrics and fault data belonging to a previous software version are used to build the software fault prediction model for the next release of the software. However, there are certain cases when previous fault data are not present. In other words predicting the fault-proneness of program modules when the fault labels for modules are unavailable is a challenging task frequently arised in the software industry There is need to develop some methods to build the software fault prediction model based on unsupervised learning which can help to predict the fault–proneness of a program modules when fault labels for modules are not present. One of the methods is use of clustering techniques. Unsupervised techniques like clustering may be used for fault prediction in software modules, more so in those cases where fault labels are not available. In this study, we propose a Machine Learningclustering based software fault prediction approach for this challenging problem.

I. INTRODUCTION

SOFTWARE quality models are useful tools toward achieving the objectives of a software quality assurance initiative. A software quality model can be used to identify program modules that are likely to be defective. Subsequently, the limited resources allocated for software quality inspection and improvement can be targeted toward only those program modules, achieving cost-effective resource utilization. A software quality estimation model allows the software development team to track and detect potential software defects relatively early on during development, which is critical to many high-assurance systems.

A software quality model is typically trained using software measurement and defect (quality) data of a previously developed release or similar project. The trained model is then applied to modules of the current project to estimate their quality. Such a supervised learning approach assumes that the development organization has experience with systems similar to the current project and that defect data are available for all program modules in the training data. In software development practice, however, various practical issues limit the availability of defect data for modules in the training data. For example, an organization may have not recorded or collected software defect data from previous releases or similar projects. In addition, since the organization may not have experience developing a similar system, the use of software measurement and defect data of previous projects for modeling purposes is inappropriate. In current times, where globalization of technology has gained momentum, distributed software development is not uncommon. Under such conditions, software defect data may not be collected by all development sites depending on the organizational structure and resources of individual sites. The absence of defect data or quality-based class labels from the training data prevents following the commonly use supervised learning approach to software quality modeling. Consequently, the task of software quality estimation or labeling program modules as fault prone (fp) or not fault prone (nfp) falls on the software engineering expert. The process of labeling each program module one at a time is a laborious, expensive, and time-consuming effort. We propose a semi supervised clustering scheme to aid the expert in the labeling process. The proposed scheme is based on constraint-based clustering using k-means as the underlying algorithm. During the k-means clustering process, the constraint maintains membership of modules (instances) to clusters that are already labeled as either fp or nfp. The proposed approach is to analyze the quality of unlabelled S/W modules using two stage approaches. One is sing metrics thresholds and another one is using FuzzyC Means clustering and then comparing both in terms of time and mean squared error value. The clustering methods create groups of

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

objects and objects in one cluster are similar whereas objects in Clustering methods can be used to group the modules having similar metrics by using similarity measures or dissimilarity measures (distances). After clustering phase, an expert or an automated approach can check the representative modules of each cluster and then, decide to label the cluster as fault-prone or not fault-prone. In this study, we show that software metrics thresholds can be used to label the clusters instead of using an expert for this time-consuming stage different cluster are dissimilar. Clustering is basically Partitional Clustering: Given a database of n objects, a partitional clustering algorithm constructs k partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the sum of squared distance from the mean within each cluster. Types of Partitional clustering: K-means Clustering and Machine Learning Clustering.

MACHINE LEARNING: In machine learning, data plays an important role, and the machine learning is used catch on and learn properties from the data. The learning and prediction performance will effect on the quantity and quality of data set.

TYPES OF MACHINE LEARNING

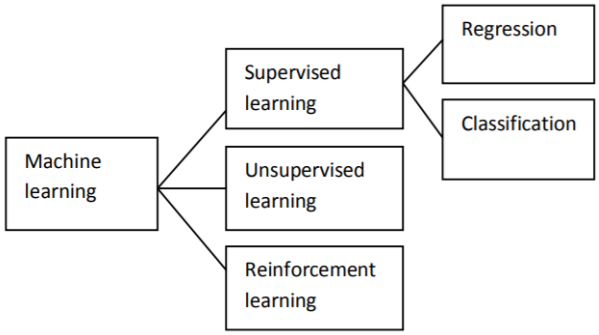

Figure 1 Types of Machine Learning

- **Supervised learning:** Supervised learning is trained a labeled data. Supervised learning generally used in applications where historical data predict like future use. This technique further divided into two categories as regression and classification. In regression the label is continuous quantity. On the other hand, in classification the label is discrete

- **Unsupervised learning:** Unsupervised learning used unlabeled data that has no historical labels.

- **Reinforcement learning:** It is used for gaming, navigation and robotics. This type of learning has three types of primary components: the learner, the environment and actions.

- **Machine learning techniques:** A machine learning algorithms are developed to build machine learning models and important machine learning process. In this paper we discuss three classifiers like decision tree, naïve bayes and support vector machine.

(1) **Decision tree:** decision tree based on supervised learning algorithm. It is one of the predictive model approaches used in machine learning, data mining and statistics. In decision analysis, a decision tree can used to usually represent decision making and decisions. In data mining, a decision tree characterize data but not decision s, to some extent the resulting allocation tree can be an input for decision making.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

There are many decision tree algorithms:

C4.5 (successor of ID3) CART (classification and regression tree)

MARS: extends decision tree to better handle numerical data.

(2) **Naïve bayes:** Naïve bayes classifier is widely studied probabilistic learning method. Naïve Bayesian classifier conclude that there are no addiction among attributes. This presumption is called conditional independence.

Advantages of naïve bayes:

- It is used a very perceptive technique.
- It widely studied expectation learning method.
- Naïve bayes classifier is computational fast when executing decision.

(3) **Support vector machine (SVM):** "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However it is mostly used in classification problems. SVM used for classification of both linear and non-linear data. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

Support vector machine some steps:

- Set up the training data
- Set up of SVM parameters
- Types of SVM kernel
- Train the SVM
- Regions classified by SVMP
- Support vectors

II. EXISTING SYSTEM

In Existing, Quad Tree-based K-Means algorithm has been applied for predicting faults in program modules. First, Quad Trees are applied for finding the initial cluster centers to be input to the K-Means Algorithm. An input threshold parameter governs the number of initial clustercenters and by varying the user can generate desired initial cluster centers. The concept of clustering gain has been used to determine the quality of clusters for evaluation of the Quad Tree-based initialization algorithm as compared to other initialization techniques. The clusters obtained by Quad Tree-based algorithm were found to have maximum gain values. Second, the Quad Treebased algorithm is applied for predicting faults in program modules.

Disadvantage in Existing System

- The K-Means algorithm is very sensitive to noise.
- The Quad Tree-based method assigns the appropriate initial cluster centers.
- Using both Quad Tree and K-Means the Overall processing time was increased.
- In software fault prediction, the value of threshold is not clearly described.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

III. PROPOSED SYSTEM

In the proposed system we propose a Machine Learning clustering technique for software fault prediction. First we select the input dataset and apply the attribute selection technique. Attribute selection technique is used to select the important attribute and reduce the number of attributes. Attribute Evaluation approach is used to evaluate the attribute and return the weight of the attribute. Then apply the ranking method for get the most weighted attributes. After the Attribute selection we get the reduced number of attributes. This reduced attribute is used to clustering approach. Given a database of n objects, a partitional clustering algorithm constructs k partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the sum of squared distance from the mean within each cluster. Fuzzy-C means is a one type of partitional clustering approach. To cluster the data points, we are using Fuzzy-C means clustering. After the clustering approach each cluster maintain the centroid value. Metric threshold is approach used to define the threshold value. Compare this metric threshold and centroid data values. If any metric value of the centroid data point of a cluster was greater than the threshold, that cluster was labelled as faulty and otherwise it was labelled as nonfaulty.

Advantage in proposed system

- Propose a single technique
- FUZZY C means is used for Clustering.
- Processing time is reduced
- Clear description of Metric Threshold

IV. SYSTEM ARCHITECTURE

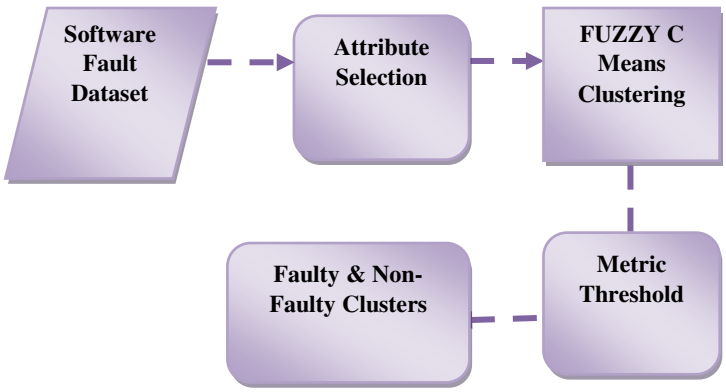

Figure 2 Proposed System

Fault Dataset: Collect Software Fault Dataset (AR3, AR4 and AR5).Read the data of the Dataset and stored into DB.Get the Attribute names for Attributes Selection.

Attribute Selection: Get all Attribute in our dataset**,** The dataset contains totally 30 numbers of attributes.Attribute Selection is the technique of selecting a subset of relevant features for building robust learning models.We take all attributes into our process it takes so much time for processing and increase the work burden**.** So, we reduce the total number of attributes and consider high relevance attributes only.Calculate the relevance of an attribute using Attribute Evaluation.We use Weka tool for attribute selection and ranker.

Machine Learning clustering: A cluster is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.Machine Learning (FCM) is a method of clustering which allows

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

one piece of data to belong to two or more clusters.The data's are initially clustered and then performing Machine Learning algorithm.After the clustering process we separate clustered data's and cluster centroid.

Metric Threshold: Determine the acceptable metrics thresholds using some parameters. The Parameters are,

- Lines of Code (LoC),
- Cyclomatic Complexity (CC),
- Unique Operator (UOp),
- Unique Operand (UOpnd),
- Total Operator (TOp),
- Total Operand (TOpnd).

Finally, we have the threshold vector [LoC, CC, UOp, UOpnd, TOp, TOpnd]

Detect Fault: After the clustering process each cluster maintain the datapoint.Get the metric threshold for each cluster.If any metric value of the centroid data point of a cluster was greater than the threshold, that cluster was labeled as faulty and otherwise it was labeled as nonfaulty

Just-in-time defect prediction aims to predict if a particular file involved in a commit (i.e., a change) is buggy or not. Traditional just-in-time defect prediction techniques typically follow the following steps:

1) Training Data Extraction. For each change, label it as buggy or clean by mining a project's revision history and issue tracking system. Buggy change means the change contains bugs (one or more), while clean change means the change has no bug.

2) Feature Extraction. Extract the values of various features from each change. Many different features have been used in past change classification studies.

3) Model Learning. Build a model by using a classification algorithm based on the labeled changes and their corresponding features.

4) Model Application. For a new change, extract the values of various features. Input these values to the learned model to predict whether the change is buggy or clean.

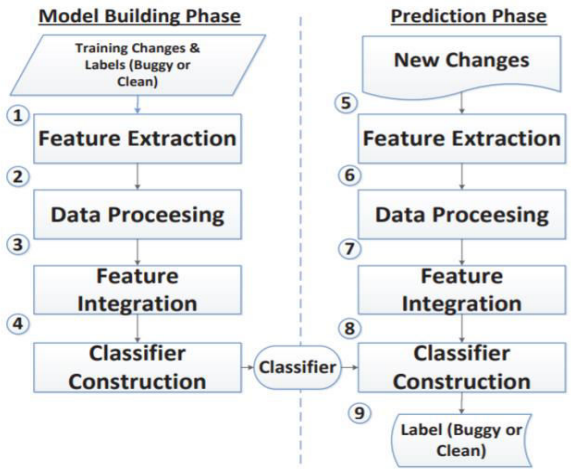

Figure 3 Flow of Proposed System

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

The framework mainly contains two phases: a model building phase and a prediction phase. In the model building phase, our goal is to build a classifier (i.e., a statistical model) by leveraging deep learning and machine learning techniques from historical changes with known labels (i.e., buggy or clean). In the prediction phase, this classifier would be used to predict if an unknown change would be buggy or clean.

Performance measures: Machine learning checks the prediction performance with the help of various performance measures:

- Precision
- Recall
- Accuracy
- F measure
- ROC (receiver operating characteristics)

There were calculated using the prediction classification confusion matrix table:

		PREDICTED	
		P	N
ACTUAL	P	TRUE POSITIVE (TP)	FALSE NEGTIVE (FN)
	N	FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)

Table 1: Confusion Matrix

TP (true positive): Number of correct predictions that an instance is positive.

TN (true negative): Number of correct predictions that an instance is negative.

FN (false negative): Number of incorrect predictions that an instance is positive.

FP (false positive): Number of incorrect predictions that instance is negative.

Accuracy: The total number of predictions that were correct

$$\text{Accuracy } (\%) = (TP+TN)/(TP+FP+FN+TN)$$

Precision: The predicted true pages those were correct:

$$\text{Precision } (\%) = TP/(TP+EP)$$

Recall: The predicted true pages that were correctly identify.

$$\text{Recall } (\%) = TP/(FN+TP)$$

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

F-Measure: Derives from precision and recall values:

FMeasure (%) = (2 x recall x precision)/ (recall + precision)

V. RESULTS AND ANALYSIS

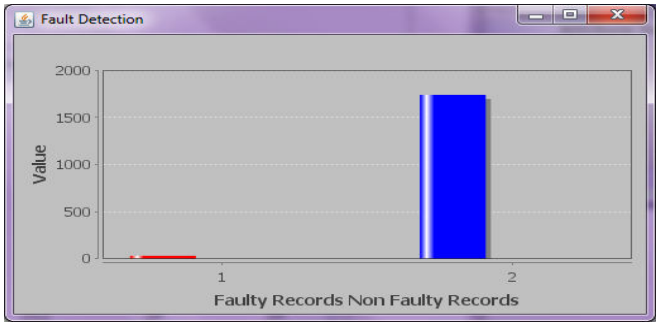

Figure 4 Fault Detection Rate

Figure 5 Error Rate

VI. CONCLUSION

We proposed a Machine Learning clustering and Metrics threshold-based software fault prediction approaches for the cases where there is no priori fault data. Attribute selection method is used to select a weighted attribute and returns a most weighted attribute only. After the clustering method the data points are clustered and each cluster have the own centroid value. Metric Threshold was used to define the threshold and it's compared to each cluster centroid. Finally, faulty and non-faulty data points was displayed.

REFERENCES

1. "Software Fault Prediction and Defect Estimation Using Machine Learning andKMedoids Algorithm" – V. Bhattacherjee and P.S. Bishnu, 2011
2. "Application of K-Medoids with kd-Tree for Software Fault Prediction" – P.S. Bishnu and V. Bhattacherjee, 2011
3. "Outlier Detection Technique Using Quad Tree" – P.S. Bishnu and V. Bhattacherjee, 2009
4. "Initializing K-Means Batch Clustering: A Critical Evaluation of Several Techniques" – D. Steinley and M.J. Brusco, 2007
5. "Unsupervised Learning Approach to Fault Prediction in Software Module" – V. Bhattacherjee and P.S. Bishnu, 2010
6. "Complexity Metrics for Analogy Based Effort Estimation" – V. Bhattacherjee, P.K. Mohanti, and S. Kumar, 2009

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 5, May 2022||

| DOI:10.15680/IJIRSET.2022.1105068 |

7. "Analyzing Software Measurement Data with Clustering Techniques" – S. Zhong, T.M. Khoshgoftaar, and N. Seliya, 2004
8. "Clustering and Metrics Threshold Based Software Fault Prediction of Unlabeled Program Modules" - C. Catal, U. Sevim, and B. Diri, 2009

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details