ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Clustering and Classification Based Prediction for the Design of Effective Complaint Systems

S Rajarajan*, M Prabhu, R Banumathi

School of Computing, SASTRA University, India

ABSTRACT

The hardware and software combination is an integral part for any testing to be done and it also enables testing to generate precise results. Nowadays the area of hardware and software is evolving and outdating so fast. So the possibility of getting faults in any electronic device is very high. It would be beneficial for the buyers to know the chances of getting complaints in any device before they actually purchase. It will make customers to avoid dilemma and will makes them to buy confidently. It increases the loyalty of brands and companies among their customers. Customer satisfaction will improve significantly. The proposed system uses the concept of clustering and classification. Clustering is the process of grouping different data into one cluster. The data which have same properties will be clustered together. After clustering the electronic complaints are split into decision classes using the decision tree. The paper generates a minimum spanning tree with correlation to the features of hardware and software electronic complaints generated. The tree is used to interpret the best possible combination for the test case in terms of hardware and software electronic complaints for the dataset.

Keywords: Complaint System, J48, Minimum Spanning Tree (MST), Agglomeration, CRM, Naïve-Bayes

***Corresponding author**

ISSN: 0975-8585

INTRODUCTION

The problem occurs in classification of software faults because of two reasons: a). when many users report so many failures for the particular software b). By running test suite it is possible to produce large number of failures[1-2]. Failures will be retrieved based on its relative features and grouped into smaller clusters based on its common defects. Identification of the group is to be done before the Failure causes are identified. It is possible that Failures that occur fall into some groups and the groups are not too large i.e. they have small count. It is necessary to identify the group first which makes it easier to identify the causes of failure. Some types of failure are very easy to classify but some types are not so easy to classify.

As software systems continue to grow in size and in the complexity, they are increasingly designed to be configurable. This is desirable because it enables systems to yield better performance. At the same time, however, configurability can greatly complicate software development tasks, such as testing, because each configuration can contain unique faults, and therefore, each configuration may need to undergo expensive testing—something that is generally infeasible in practice. There are many multi-variants Data Mining and Data Analysis algorithms or techniques are present which can simplify the classification problem. These techniques should be applied to execution profile. The requirements of this approach can be categorised into three types of information about the execution being processed and checked. This technique (execution profiles) shows or reflects the cause of the defects. Inspecting the information that can be used to confirm the reported failures and the technique also diagnose the failure which gives the appropriate result. There are other methods available to classify the electronic complaints.

Cluster Analysis

It is the process of classifying the entities into many groups which implies the dataset is being partitioned or divided into different subsets. These subsets are called clusters. Data in same cluster have the common properties in some ways. The primary goal of data clustering is grouping the data into clusters in such a way that data available in same cluster will be more similar than the data which are present in different clusters. It organizes relevant patterns into clusters.

Data Set Clustering

Data clustering algorithms is technique which is dependent on singe term analysis. There are many types of data clustering techniques among which the best technique is hierarchical data clustering. This data clustering technique organizes the data into different levels. The root is the most abstract data which consists the most part information but shows very less. The hierarchical data clustering is having two parts. First part is data index model and the second part is the improvement model.

RELATED WORKS

Many prediction methods have been developed so far. But all those methods do not get the correct prediction and sometimes they have the probability of 0.5 to get wrong prediction. Bagging prediction method is a method which is used to produce multiple versions of different predictors which in turn can be integrated to get a better result [3]. In [4], they propose the technique which finds the assets or properties of a program which consist of the errors. When a programmer writes a program it surely consists of some errors to detect the errors, the tester runs test suite on test cases and if all the test cases pass the test it is considered that there is no further error present in the program. But there can be the case when still errors are present which are hidden from the programmer. This paper presents the techniques for finding these hidden errors. The method used by them is decision tree algorithm, support vector learning machine algorithm. Many testing concepts have been developed but no concept is capable of handling electronic complaints. The approach of [5] presents the concept which can also be applied on the electronic complaints. It uses the concept of integration testing which provides a general way testing. It is the testing which can be applied on both the hardware and software. There are many number of ways in that a system must be tested can often be overwhelming. There are number of tools which can be used to generate automatic test cases, but these tools have disadvantage of combinatorial explosions many test cases. In [6], the authors have combined table based testing and code coverage with Bellcore's Automatic Efficient Test case Generator (AETG) to generate small efficient sets of test cases. AETG is used to generate the tables test vectors by using pair-wise test case

generation technique , which can be executed using test case driver immediately. Then Code coverage technique is used to find missing functionality from AETG's model. The first trial was taken at the Nortel email where they were able to cove around 97% test cases. [7] is an approach for context-aware public displays to improve personalized information access according to a user's language, location, time or other individual preferences using data mining The modern computer world is emerging rapidly so the chances of the errors also increasing rapidly. It is impossible to think all the test cases and check one by one. This paper present a new execution tool KLEE which is capable of producing test cases automatically. It is checked on a program which had 89 test cases and all the test cases were covered by this tool. So the result of this tool is very good and it is very fast also.

Many testing concepts have been developed but no concept is capable of handling electronic complaints. This paper presents the concept which can also be applied on the electronic complaints. It uses the concept of integration testing which provides a general way testing. It is the testing which can be applied on both the hardware and software. The data is collected from online portals and the methodologies are applied.

PROPOSED METHODOLOGY

Architecture

The data set comprise of hardware and software configuration which will use clustering technique to cluster the data in a way that data with same structure are clustered together. Those clustered data pass through decision tree algorithm which produces the tree. Latter the tree is pruned to give the specific rules to aid predictions. Fig. 1 presents the proposed architecture.

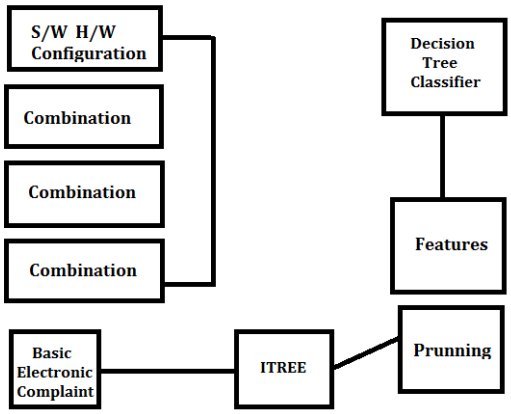

Fig 1: Proposed System's Architecture

CLUSTER GENERATION

Distance Measure

This is the important step which enables us to know whether elements have similarity or not and if similarity exists it will be calculated. Based on the distance existing between two elements (may be the elements are close to each other or it may not) it have its impact on the shape of the cluster. Let take the example of 2D space in which the distance between the two points (1,0) and the origin point(0,0) is always one based on usual norms. But in the case of other norms distance between those points will give result as 2,$\sqrt{2}$ or 1.

Finding the Distance

The Euclidean distance is measured by taking the distance exists between two points are so called in Euclidean space. Here Euclidean distance is used as the metrics. In our project we use the Euclidean distance to measure the distance between two points in same cluster. Points which are having same distance is considered as the same in properties is some way.

ISSN: 0975-8585

Clusters Creation

Hierarchical clustering begins with the making of clusters hierarchy. In earlier method they used tree structure for the hierarchy representation. Agglomerative algorithms and decisive algorithms differs in such a way that in earlier case it will start from the beginning of the tree which means it will start from the leaves and in latter case it will start from the ending which means it will start from the root of the tree.

Agglomeration

For example, the given data has to be clustered and distance metric is to be taken. Fig.2. shows the process.

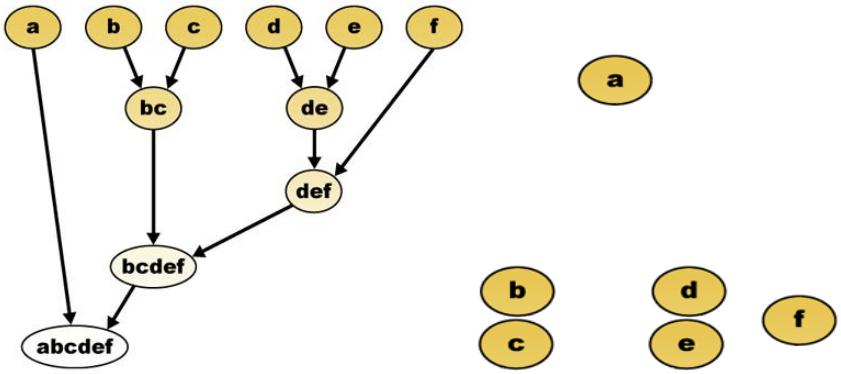

Fig 2: Data Clustering -Raw data and Resultant Diagram

The following formula is used for the processing. A and B are the nodes and d(x, y) is the distance exists between two nodes x and y.

$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y)$$

Data Set Preparation and Loading for Electronic Complaints

The system datasets of the various software's and considered architectures are first collated in this module. It consists of the components such as one is about removal of the irrelevant features and the other feature is about the elimination of the redundant features. The irrelevant features are normally eliminated by examining the features which are close to the target and the redundancy is eliminated by picking the representative feature from different clusters. The final subset will be obtained as result. The model considers the first n features of the configuration file in the UNIX file system.

The configuration space considered to be effective present in the system will be usually smaller than its full configuration space. . To illustrate why this might be true, consider a program with four options: a, b, c, and d(all are binary valued).consider the assumption taken that all 16possible electronic complaints of these options are valid. Assume also that for the system's test suite and with the testing goal of 100 percent, resulting in three main interactions.

All three interactions are satisfied by a single concrete configuration, namely a ^ b ^ :c ^ d. Thus, for this system, coverage goal, and test suite, the effective configuration space contains only one configuration, while the full configuration space has 16. Moreover, since at least one of the system's interactions involves three options, covering arrays of strength 2 or less would not be guaranteed to achieve maximal coverage, while covering arrays of strength 3 or higher will contain electronic complaints that add nothing to overall coverage.

Clustering

ISSN: 0975-8585

As mentioned above to do hierarchical clustering the all N number of elements to be taken and the following steps to be followed.

Step1: Each item is assigned individually to the every cluster. Consider like if there are N number of items which are present, then there will be the existence of N clusters with each data items assigned individually assigned to them.
Step2: Then in this step all clusters are compared and the clusters which are having similarity are grouped together and one cluster is reduced.
Step3: This step calculates the similarity between the newly formed cluster and the old cluster..
Step 4 : The main goal of this step is to form a single cluster finally. So the step mentioned is to be repeated until the single cluster is formed. The new cluster contains the N number of items.

Third step is performed using three linkages: In the first minimum distance is consideration. Second is to consider the average distance. Third is to consider the maximum distance

Decision Trees – J48– Classifier

This algorithm is based on MST (Minimum Spanning Tree). This approach is based on the assumptions that clustering algorithms do not take data points which are grouped around centres or separated by the means of geometric curve. The algorithm works with the process that MST is to be partitioned and the representative feature is to be selected. The next step is to apply the relevant correlation measures so the irrelevant features will be removed. After the irrelevant features are removed then MST is concerned with the two components which are connected. It consists of the components such as one is about removal of the irrelevant features and the other feature is about the elimination of the redundant features. The irrelevant features are normally eliminated by examining the features which are close to the target and the redundancy is eliminated by picking the representative feature from different clusters. The final subset will be obtained as result The algorithm runs will not be terminated until it meets its stopping criteria given by the developer (e.g., time limit is getting exceeded).This work starts with the construction of interaction tree having only one true node. Once J48 starts progressing, it also keep track of records of so far execution of electronic complaints and its relevant information. Once iteration is started it finds the Best leaf node which picks the next level leaf node based on some heuristics. As the fully explored interaction tree is not expected, these heuristics are considered to be necessary.

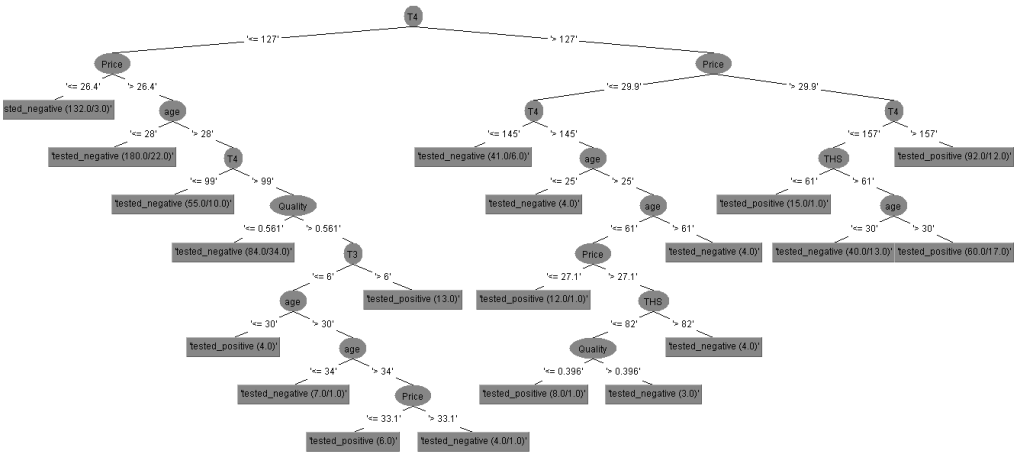

Fig 3: Graphical view of Tree

Construct Minimum Spanning Tree – Tree Discovery

This approach is based on the assumptions that clustering algorithms do not take data points which are grouped around centres or separated by the means of geometric curve. In this module, the MST is constructed from the relevant features from the dataset by eliminating the irrelevant features. In this relevant features next take a pair of features and apply feature correlation metric in between these features. This

ISSN: 0975-8585

calculates the distance and thereby the information gain using the entropy technique. After calculate the information gain of the pair of features then choose high information gain feature other features are removed from the relevant ones. As a result we get the highly correlated features separately with target concept. Now construct the MST from these relevant features using the decision tree algorithm.

Tree Partition and Building

The tree partition or clustering module is used to eliminate the redundant features. After the minimum spanning tree is constructed, the features are divided into clusters are made with the features available by using graph-theoretic clustering method. This is also called tree partitioning [8]. The tree will be partitioned below constraint. The constraint is taking a pair of features in minimum spanning tree. Calculate the information gain of these features. If the information gains of these features is less than the target concept then partition this group of features. Mostly all features will be present in the cluster. Every cluster is independently treated and taken as individual feature [9]. Next the node in the tree is used to select the representative feature in each and every cluster. We select the representative feature from each cluster based on information gain. It means choose the high information gain feature from each cluster. Finally we get highly correlated feature subset from high dimensional data. From this one can infer the correct design electronic complaint. Fig. 4 shows the tree partition that was made.

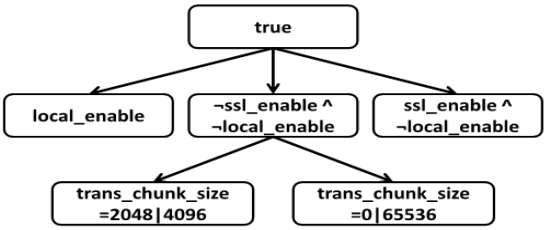

Fig 4: Tree Partition

Implementation and Results

To evaluate our proposed approach and test its efficiency in prediction, we have implemented a tool using Java. For incorporating J48 into the tool, we included the weka package. The tool is fed with the features of the products that are being evaluated and it produces the prediction about the possibility of complaints that could result in after their purchase. The fig. 5. represents the screen shot of the implementation in Java using Netbeans. For evaluating our approach we gathered sample dataset from UCI repository, which is a widely used repository of datasets for aiding machine learning activities [10]. Our evaluation of the implemented tool based on the datasets produced better results that are superior than some existing approaches. It was ascertained through the assessment that the proposed approach is cost effective, simple and efficient. We have presented the screen shot of our implementation in Fig. 5.

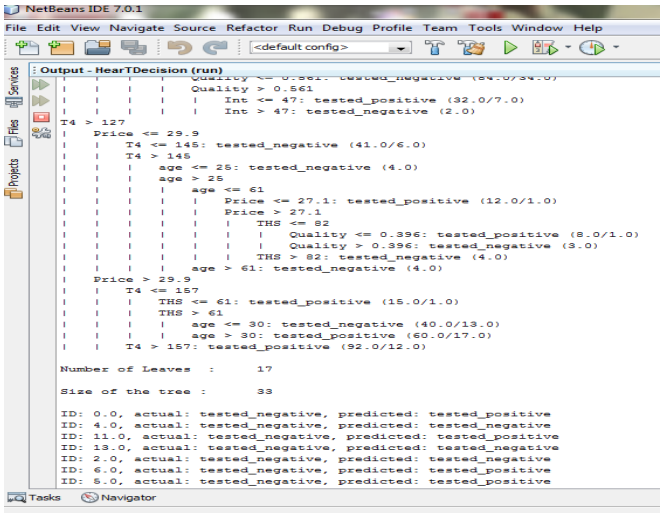

Fig 5: Screen-Shot of Implementation

CONCLUSION

As the world has migrated to a global economy model, customers have plenty of options in buying their desired products. The wide variety of choices available has made it very difficult to make their selections. Though the price and brand value are useful, the differences based on them are diminishing. When multiple products with almost similar prices and company reputations are offered, customers need some other measure to accomplish their selection. Classify products based on their chances of getting faulty based on the past usage and complaints history plays an important role. In this direction, we have implemented a new approach by using J48 algorithm for rule generation and Naïve bayes algorithm for classification. We have implemented and tested our proposed methodology by utilizing the sample data sets collected from UCI. Our tests prove that the proposed approach produces accurate results and could significantly improve customer satisfaction and company goodwill.

REFEERENCES

[1] Haran, M., Karr, A., Last, M., Orso, A., Porter, A. A., Sanil, A., & Fouche, S. (2007). Techniques for classifying executions of deployed software to support software engineering tasks. *IEEE Transactions on Software Engineering*, *33*(5), 287-304.

[2] Fouché, S., Cohen, M. B., & Porter, A. (2009, July). Incremental covering array failure characterization in large configuration spaces. In *Proceedings of the eighteenth international symposium on Software testing and analysis* (pp. 177-188). ACM.

[3] Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

[4] Brun, Y., & Ernst, M. D. (2004, May). Finding latent code errors via machine learning over program executions. In *Proceedings of the 26th International Conference on Software Engineering* (pp. 480-490). IEEE Computer Society.

[5] Bryce, R. C., & Colbourn, C. J. (2006). Prioritized interaction testing for pair-wise coverage with seeding and constraints. *Information and Software Technology*, *48*(10), 960-970.

[6] Burr, K., & Young, W. (1998, October). Combinatorial test techniques: Table-based automation, test generation and code coverage. In *Proc. of the Intl. Conf. on Software Testing Analysis & Review*.

[7] Dickinson, W., Leon, D., & Podgurski, A. (2001, July). Finding failures by cluster analysis of execution profiles. In *Proceedings of the 23rd international conference on Software engineering* (pp. 339-348). IEEE Computer Society.

[8] Francis, P., Leon, D., Minch, M., & Podgurski, A. (2004, November). Tree-based methods for classifying software failures. In *Software Reliability Engineering, 2004. ISSRE 2004. 15th International Symposium on* (pp. 451-462). IEEE.

[9] Cohen, D. M., Dalal, S. R., Fredman, M. L., & Patton, G. C. (1997). The AETG system: An approach to testing based on combinatorial design. *IEEE Transactions on Software Engineering*, *23*(7), 437-444.

[10] UCI Machine Learning Repository - http://archive.ics.uci.edu/ml/