Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# An Approach to Detect and Classify Bugs using Data Mining Techniques

**Maninderjit Kaur** *
M.Tech.,Research Scholar
CSE, RIMT-IET,Mandi Gobindgarh,
India

**Dr. Sushil Kumar Garg**
Principal
RIMT-MAEC, Mandi Gobindgarh,
India

***Abstract— Web Applications are very popular in daily life. With the rapid growth of technologies the usage of web applications are also increasing. Different languages are used for developing the web applications like C++, C#, ASP.NET, PHP, Java. The source code contains several types of bugs and errors. Software product shows some minor bugs after being released. So the products undergo Maintenance. Finding the bugs or errors is not an easy task. To find the errors deep understanding of the language is required. Manual Reviews are time consuming. Tools are used to check the accuracy of the language quickly. Data mining techniques, Clustering and Classifications are used to mine the data and extracting the meaningful and useful information. The aim of this paper is to present an approach to detect the bugs or errors in the web-based applications and to cluster and classify them for knowledge discovery.***

***Keywords— Web Application, Bug, Error, Clustering, Classification, K-Means Clustering Algorithm, CART.***

## I. INTRODUCTION

A Web Application is a type of software that is hosted on server and can be accessed remotely by a human through an Internet Browser. Other types of software are also widely used and distributed. But web-based application faces some additional challenges like maintaining a consumer base and ensuring acceptability of the application. One hour of downtime of any web application leads to huge loss for company. One solution to avoid these monetary losses and maintaining a consumer base is to design a web-based application having high security, reliability, usability, acceptability etc. A web-based application should be well-designed and well-tested so that is it is free from errors and bugs [13].

To satisfy the growth of technologies large number of web applications have been developed and deployed. A web-based application may be visited by several millions of users concurrently. Testing and performance evaluation of the web application is very important. The profile of web-based applications changes very fast. Maintenance of website is faster than other applications. Regression testing of web-based applications is important to handle such small, incremental changes. Web applications lacking significant checking of language dependent accuracy and testing. There are many reasons for this lack like short delivery time, pressure to change, developer turnover and changes in user requirements.

Bug or Defect or Fault is an incorrect process, step or data definition in a computer program. An error is difference between the desired and actual performance and behaviour of a system or object. Failure is the inability of a system to perform its required functions within specified performance requirements [9], [21]. Bugs or Errors in software can have devastating effects in today's world. Software shows some minor bugs after being released. Bugs and errors are very hard to find. To find the new error types a deep understanding of the programming language is required. Only the programmers have good knowledge of the language. Manual reviews of the code are very time consuming. As the size and complexity of software increases manual inspection becomes harder task. Software companies invest a large amount of significant resources and manpower to detect the errors and bugs in the product. Software developers and testers often work with different tools. There are many types of bugs or errors present in the code like syntax errors, queries syntax, memory usage etc.

This paper is organized in V sections. Section II describes the overview of some data mining techniques, Section III explains related work. Methodology is given in Section IV. Section V describes the conclusion.

## II. DATA MINING TECHNIQUES: OVERVIEW

Data Mining is the process of finding a small set of precious information and patterns from large sets of raw material. Data Mining has various techniques such as Frequent Pattern Mining, Pattern Matching, Clustering and Classification. Here we describe some of them:

### *A. Clustering*

Clustering is considered to be one of the most significant techniques and broad fields of data mining which is applicable in various domains such as life sciences, medical sciences, engineering and so on. Clustering technique can be

 

viewed in various perspectives depending on operational environment which includes unsupervised learning in pattern recognition, numerical taxonomy in biology and typology in social sciences and partition in graph theory etc. [22].

In clustering data elements having similarities are placed in respective groups [3]. The clusters formed as a result of clustering can be defined as a set of like elements. But the elements from different clusters are not alike. Clustering process is similar to database segmentation, where like tuples in a database are grouped together as in [19]. A good clustering method will produce high quality clusters with low inter-cluster similarity and high intra-cluster similarity [12].

The main advantage of clustering is that interesting patterns and structures can be found directly from very large data sets with not any prior knowledge about the clusters. The quality of clustering method depends on the following:

- The use of similarity measures and its Implementation.
- Efficiency of method to discover some or all of the hidden patterns.
- How to define and represent the chosen cluster [11], [12].

There are many clustering algorithms: Hierarchical Clustering, K-Means Algorithm, Density-based Algorithms.

### *B. Classification*

Classification is widely used technique in the data mining domain. For large databases scalability and efficiency are the immediate problems in classification algorithms. Classification is a supervised learning technique in data mining where training data is given to classifier that builds classification rules. Test data, is given to classifier, and then for unknown classes it will predicts values as in [15]. There are three phases of classification technique, a learning phase, a testing phase and an application phase [4].

The classification problem can be defined as follows for a database with a number of records and for a set of classes such that each record belongs to one of the given classes , the problem of classification is to decide the class to which given record belongs. Classification is one of the most important data mining techniques .It is used to predict group/class membership for data instances [25]. The goals of Classification as in [10] are:-

- Apply the model to predict the class of previously unseen records (class should be predicted as accurately as possible).
- Carry out deployment based on the model (e.g. implement more profitable marketing strategies).

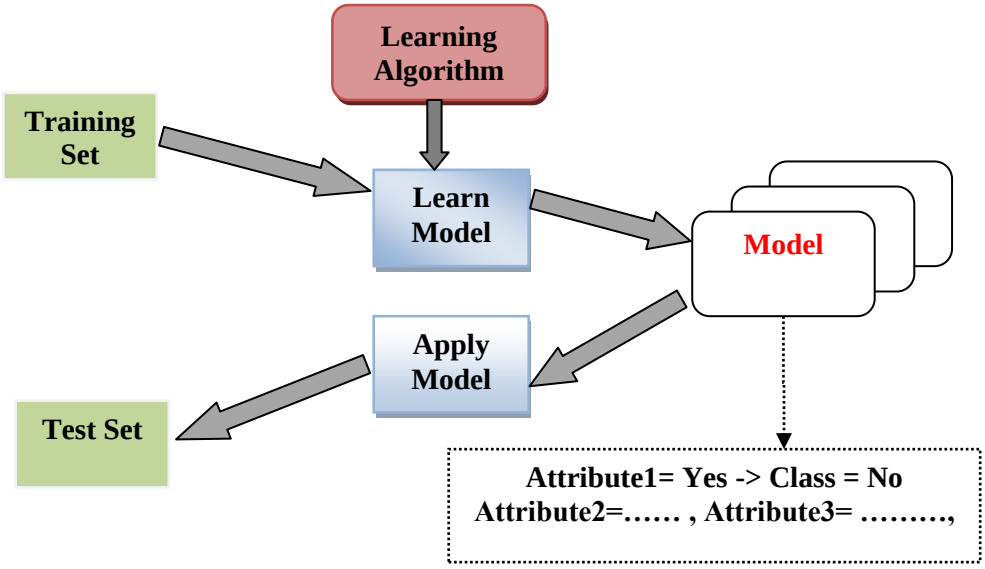


Fig. 1 Classification Task [10]

Different models have been proposed for classification such as Decision tree, Neural networks, Bayesian belief networks, Fuzzy set and Genetic models. The decision trees classifier is most widely used in among of these models for classification. They are popular because they are practically and easy to understand. Many algorithms such as ID3, C4.5 and CART (Classification and Regression Tree) have been devised for decision tree construction. All of these algorithms are used in various areas such as image recognition, medical diagnosis credit rating of loan applicants, scientific tests, fraud detection and target marketing. The decision tree is a supervised classification approach .A decision tree is a flow chart like structure, where each internal node denotes a test on an attribute, each branch shows an outcome of the test and each leaf node holds a class label. The top node in a tree is defining a root node. A decision tree has two different sub sets – a training set and a test set. The training set is used for deriving the classifier and test set is used to measure the accuracy of the classifier. The percentage of the test data set that is correctly classified is used to determine the accuracy of classifier [25]. The following table shows the comparison between the different classification algorithms [2].

*1) Advantages of Decision Trees*: In [10], the advantages of decision trees are given as follows:

- Fast at classifying unknown records.
- Small-sized trees are easily interpreted.

 

- Both continuous and discrete attributes can be handled.
- In the presence of redundant attributes work well.
- Robust to the effect of outliers.
- The fields which are most important for prediction are clearly indicated.

2) *Disadvantages of Decision Trees:* The disadvantages of decision trees are as following as in [10]:
- The construction of a decision tree can be affected by irrelevant attributes.
- Decision boundaries are rectilinear
- Very different looking trees are generated if there is a small variation in data.
- A sub-tree replication possible for several times
- Error-prone with too many classes
- Prediction of the value of a continuous class attribute is not good.

TABLE I
PARAMETER COMPARISON OF DECISION TREE ALGORITHMS

| **Algorithm** | **Parameters** | | |
|---|---|---|---|
| | **Measure** | **Procedure** | **Pruning** |
| ID3 | Entropy info-gain | Top-Down Decision Tree Construction | Pre-Pruning using a Single Pass Algorithm |
| CART | Gini Diversity Index | Construct Binary Decision Tree | Post-Pruning based on Cost- Complexity Measure |
| C4.5 | Entropy info-gain | Top-Down Decision Tree Construction | Pre-Pruning using a Single Pass Algorithm |
| SLIQ | Gini Index | Decision Tree Construction in a Breadth First Manner | Post-Pruning based on MDL Principle |
| SPRINT | Gini Index | Decision Tree Construction in a Breadth First Manner | Post-Pruning based on MDL Principle |

## III. RELATED WORK

In [16], in this research, they discuss classification, clustering and association mining for defect prediction. These techniques help the developers to detect and correct the defects. Results are compared at end to see which is better.

In [20], this paper investigates multiple feature selection techniques that are generally applicable to classification-based bug prediction methods. Less important features are discarded until optimal classification performance is reached. The features used for training is reduced, often to less than 10 percent of the original features. The performance of Naive Bayes and Support Vector Machine (SVM) classifiers is characterized on 11 software projects by using this technique. The process is fast performing. It can be applied to predicting bugs on other projects.

V.Neelima et al. [24] uses Text Mining Techniques for bug detection. In order to alleviate the overhead in debugging, they proposed an approach to detect bugs in C programs via matching and mining techniques. The input to the system is the text file containing syntax errors, matched with database that acts as a repository, classify them and generate the analysis report that gives solution. The scope of this work is that other types of errors can be considered and can be used for other programming languages.

In [7], this study aims to develop two-phase prediction model that uses bug reports content to suggest the files to be fixed. The first phase of model checks whether the given bug report contains sufficient information for prediction. If so, on the basis of content of the bug report, the model proceeds to predict files to be fixed. Comparison is done with other models. They use Naïve Bayes Classification algorithm. If approach fixed a location, (for almost half of the bug reports) 70 percent of the recommendations point to correct files.

In [17], the proposed system analyze the software defects, categorize them using clustering approach then defects are measured in each cluster separately. In order to improve quality of software development, they make use of Data mining Clustering technique. This paper reviewed the software defect management based on different types of defects by using clustering algorithms. The resulting data is used as the basis for determining the nature of defects.

Priyank Dineshkumar Patel [18], presents defect forecasting algorithms that analyze a software project's change history. Defects are related. Their algorithm finds this relation by storing locations that are likely to have defects and useful to find most defect prone files. Open source projects with more than 5,000 revisions are evaluated. The result shows that the selected defect training data account for 72%-90% of future defects. This information can be used to increase software quality and reduce software development cost. CVSAnalY tool is used for extracting revisions from repositories. It is easy to understand, implement and integrate in live projects. It's also concluded by this research that accuracy of future defects forecasting is increasing if few history available of software system.

 

## IV. METHODOLOGY

In Fig. 2 the Proposed System's overview is given. There are three components of the system: Input, Processing and Display. Source code or source code files are the inputs to the system. The detection of bugs or errors will be done in this code. After the detection of the bugs the clustering and classification of the detected bugs will be performed. Classification is used for the knowledge discovery. After that the results will be displayed and compared to the results of existing work.

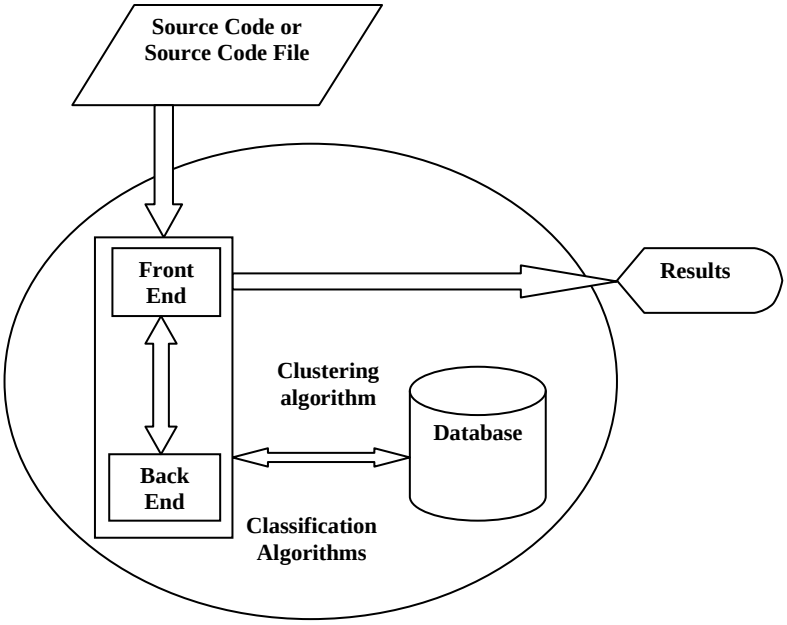


Fig. 2 Overview of the System [24]

### *A. Outline of Algorithms*

The present work uses Modified K-Means Algorithm for Clustering and CART algorithm for Classification. The overviews of the algorithms are as follows:

1) *K-Means Algorithm*: K-Means is one of the simplest unsupervised learning methods among all partitioning based clustering methods. It classifies a given set of n data objects in k clusters, where k is the number of desired clusters and it is required in advance [11].

**Input**: 'k', the number of clusters to be partitioned; 'n', the number of objects.

**Output**: A set of 'k' clusters based on given similarity function.

**Steps**:

**i)** Arbitrarily choose 'k' objects as the initial cluster centres;
**ii)** Repeat,

a. (Re) assign each object to the cluster to which the object is the most similar; based on the given similarity function;
b. Update the centroid (cluster means), i.e., for each cluster calculate the mean value of the objects;

**iii)** Until no change.

**Advantages:** K-mean algorithm is a classic algorithm to resolve cluster problems; this algorithm is relatively simple and fast. This algorithm is relatively flexible and high efficient for large data collection, because the Complexity is O (ntk). Among which, n is the times of iteration, k is the number of cluster, t is the times of iteration. It has many other advantages.

**Disadvantages:** The K value is most important for K-means clustering algorithm. K-means clustering algorithm has a strong sensitivity to the noise data objects. K-means clustering algorithm has many limitations on amount of data. In the iterative process, every time you need to adjust the cluster to which data object belongs and compute cluster centre, this algorithm is not applicable for large amount of data [5].

There is a need to modify the K-Means clustering algorithm to resolve the disadvantages of the algorithm. A large number of modifications are done to improve the working of this. In our approach we use the modified K-Mean algorithm so that some of the disadvantages will be removed like no repetition in the allocation of memory for the clusters.

 

2) *CART Algorithm:* CART (Classification and Regression Trees) was introduced by Leo Brieman et al. in 1984. CART is a binary recursive partitioning procedure. CART adopts greedy approach in which decision trees are constructed in top-down, recursive, divide-and-conquer manner. It is implemented serially. It is capable of processing both continuous and nominal attributes both as targets and predictors. Trees are grown to their maximum size. After that cost-complexity pruning is done. The "right sized" and "honest" tree is identified. It removes unreliable branches from decision tree to improve accuracy. It includes automatic class balancing, missing value handling, and allows cost-sensitive learning as in [26]. In CART decision trees are formed by a collection of rules based on variables in data set. The algorithm is as follows [6]:

**i)** Rules based on variables values are selected to get the best split to differentiate observations based on the dependent variables.

**ii)** Once a rule is selected and split a node into two, the same process is applied to each "child" node (i.e. it is a recursive procedure).

**iii)** Splitting stops when CART detects no further gain can be made or some pre-set splitting rules are met (data is split as much as possible and then the tree is later pruned).

In [23], Classification tree is based on binary splitting of the attributes. It uses Gini Index to select splitting attributes. Gini Index is defined as:

$$Gini\ (T) = 1 - \sum_{j=1}^{n} P_j^2 \qquad [23]$$

Splitting rules of CART are always couched in the form as following:

*An instance goes left if CONDITION, and goes right otherwise*

For continuous attributes CONDITION is expressed as "attribute $X_i$ <=C" and for nominal attributes as membership in an explicit list of values. CART authors argue that binary splits are preferred because (i) they fragment the data more slowly than multi-way splits, and (ii) repeated splits on the same attribute are allowed [26], [14]. It is a data exploration and prediction algorithm. It is classification method that uses historical data to construct the decision tress. Then decision trees are used to classify the new data. Number of classes must be known in advance to use CART [1]. Classification tree have classes and regression trees don't have classes [8].

### *B. Flow Chart*

Fig. 3 shows the flow chart of the present approach.

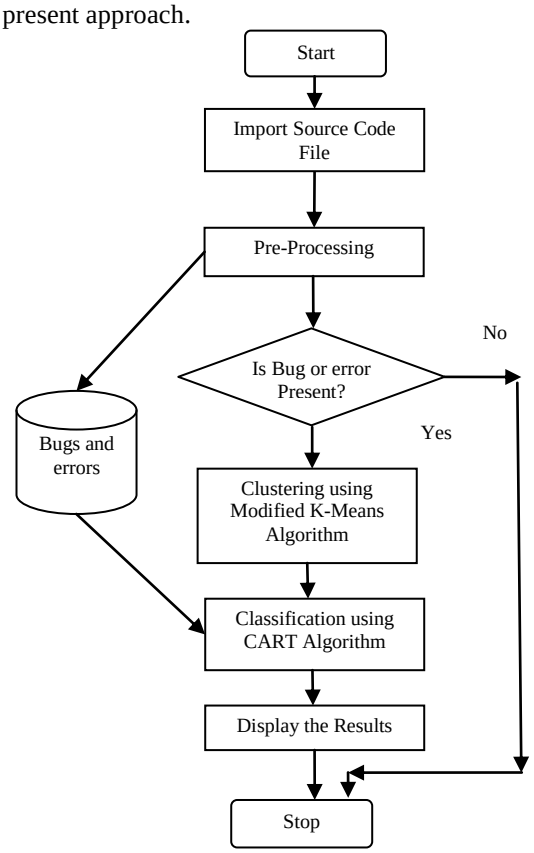


Fig. 3 Flow Chart

 

## V. CONCLUSIONS

In this paper, overview of data mining techniques such as clustering and classification is given. From the study, concluded that manual review of the source code to detect the errors or bugs is a time consuming process. Web-based applications are undergo maintenance very quickly. So there is a change occurs in the source code. To detect the errors and bugs in the large source code is very hard. We present an approach in which the bugs or errors are detected in web-based applications. The methodology of the given work uses the modified K-Means algorithm and CART classification algorithm to classify the detected bugs or errors.

## ACKNOWLEDGMENT

The author would like to thank the RIMT Institutes, Mandi Gobindgarh-147301, Fatehgarh Sahib, Punjab, India. Author would also wish to thank editors and reviewers for their valuable suggestions and constructive comments that help in bringing out the useful information and improve the content of paper.

## REFERENCES

[1] Aman Kumar Sharma, and Suruchi Sahni, "A Comparative Study of Classification Algorithms for Spam Email Data Analysis" ISSN : 0975-3397 International Journal on Computer Science and Engineering (IJCSE) ,Vol. 3 No. 5, May 2011.

[2] Anuja Priyama, Abhijeet, Rahul Gupta, Anju Rathee, and Saurabh Srivastava, "Comparative Analysis of Decision Tree Classification Algorithms" ISSN: 2277 – 4106 International Journal of Current Engineering and Technology , Vol.3, No.2, June 2013.

[3] Azhar Rauf, Sheeba, Saeed Mahfooz, Shah Khusro and Huma Javed, "Enhanced K-Mean Clustering Algorithm to Reduce Number of Iterations and Time Complexity" ISSN: 1990-9233 Middle-East Journal of Scientific Research 12 (7): 959-963, 2012.

[4] B V Chowdary, Annapurna Gummadi, UNPG Raju, B Anuradha and Ravindra Changala, "Decision Tree Induction Approach for Data Classification Using Peano Count Trees" ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Vol.2,Issue 4,April 2012.

[5] Chunfei Zhang, and Zhiyi Fang, "An Improved K-means Clustering Algori" Journal of Information & Computational Science 10: 1 (2013) 193–199.

[6] David G.T. Dension, Bani K. Mallick, Adrian F.M. Smith, "*A Bayesian CART Algorithm*", Biometrika, Vol. 85, No. 2 (Jun., 1998), 363-377.

[7] Dongsun Kim, Yida Tao, Sunghun Kim, Andreas Zeller, "Where Should We Fix This Bug?A Two-Phase Recommendation Model," IEEE Transaction on Software Engineering, Vol.39, No.11, November 2013, IEEE Computer Society 2013.

[8] Hardeep Kaur, and Harpreet Kaur, " Proposed Work for Classification and Selection of Best Saving Service for Banking Using Decision tree Algorithms" ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Vol.3,Issue 9,September 2013.

[9] http://en.wikipedia.org/wiki/Error

[10] http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec3.pdf

[11] Kahkashan Kouser, and Sunita, "A comparative study of K Means Algorithm by Different Distance Measures" ISSN (Print): 2320-9798|ISSN (Online):2320-9801 International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) Vol.1,Issue 9, November 2013.

[12] Khaled W. Alnaji, and Wesam M. Ashour, "A Novel Clustering Algorithm using K-means (CUK)" International Journal of Computer Applications (0975 – 8887) Volume 25– No.1, July 2011.

[13] Kinga Dobolyi, "An Exploration of User-Visible Errors in Web-based applications to Improve Web-based Applications Ph.d. Dissertation School of Engineering and Applied Science, University of Virginia, May 2010.

[14] Matthew N. Anyanwu, and Sajjan G. Shiva, "Comparative Analysis of Serial Decision Tree Classification Algorithms" International Journal of Computer Science and Security, (IJCSS) Volume (3): Issue (3).

[15] Mohd. Mahmood Ali, Mohd. S. Qaseem, Lakshmi Rajamani, A. Govardhan, " EXTRACTING USEFUL RULES THROUGH IMPROVED DECISION TREE INDUCTION USING INFORMATION ENTROPY" International Journal of Information Sciences and Techniques (IJIST) Vol.3, No.1, January 2013 DOI.

[16] Ms. Puneet Jai Kaur, Ms. Pallavi, "Data Mining Techniques for Software Defect Prediction ISSN (Print): 2279-0063|ISSN (Online):2279-0071 International Journal of Software and Web Sciences 3(1), Dec. 2012-Feb. 2013, pp. 54-57.

[17] P. V. Ingle, M. M. Deshpande, "Software Quality Analysis with Clustering Methods",ISSN 2249-0868 International Journal of Applied Information Systems (IJAIS), Foundation of Computer Science (FCS) New York, USA and International Conference and Workshop on Addvanced Computing (ICWAC), 2013.

[18] Priyank Dineshkumar Patel, "Defect Forecasting in Software System – Mining Approach" ISSN: 2319-7242 International Journal of Engineering and Computer Science (IJAECS) Vol.3, Issue 1, pp. 3763-3767, January 2014.

[19] S. Revathi, and Dr.T.Nalini, "Performance Comparison of Various Clustering Algorithm" ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Vol.3,Issue 2,February 2013.

 

[20] Shivaji et. al.," Reducing Features to Improve Code Change-Based Bug Prediction" IEEE Transactions on Software Engineering, Vol. 39, No. 4, April 2013, IEEE Computer Society 2013.

[21] Shivkumar Hasmukhrai Trivedi, "Software Testing Techniques" ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Vol.2, Issue 10, October 2012.

[22] Suma. V, Pushpavathi T.P, and Ramaswamy.V, "An Approach to Predict Software Project Success by Data Mining Clustering," International Conference on Data Mining and Computer Engineering (ICDMCE'2012) December 2012.

[23] T.Miranda Lakshmi, A.Martin, R.Mumtaj Begum, and Dr.V.Prasanna Venkatesan, " An Analysis on Performance of Decision Tree Algorithms using Student's Qualitative Data" I.J.Modern Education and Computer Science, 2013, 5, June 2013, pp.18-27.

[24] V.Neelima, Annapurna.N, V.Alekhya, and Dr.B.M.Vidyapathi, "Bug Detection through Text Data Mining" ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Vol.3,Issue 5,May 2013.

[25] Varsha Choudhary, and Pranita Jain, " Classification: A Decision Tree For Uncertain Data Using CDF" ISSN: 2248-9622 International Journal of Engineering Research and Applications (IJERA) Vol. 3, Issue 1, January - February 2013, pp.1501-1506.

[26] XindongWu , Vipin Kumar , J. Ross Quinlan , Joydeep Ghosh , Qiang Yang, Hiroshi Motoda , Geoffrey J. McLachlan , Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou , Michael Steinbach, David J. Hand , Dan Steinberg, "Top 10 algorithms in data mining" Springer-Verlag London Limited 2007.

 