Survey of Risk in Software Project Development by Data Mining Tree Algorithms

Dhyan Chandra Yadav, Saurabh Pal

Abstract— The knowledge or technical skill have major role in software engineering. Correction and completion are necessary part in software coding and documentation. Every software has different quality according to the requirement of the software project. Mistaken bug is detected by software tracking system. Mistaken bug problem is a coverable problem and mentation in the problem report for correction. In this paper we proposed and analyzed three data mining tree algorithms find the geometric mean, J static coefficient parameter (for sensitivity and specificity),correctness, completeness, error rate of data object by the help of AD Tree, Lad Tree and REP Tree algorithms.

Index Terms— Classification: AD Tree, LAD Tree and REP Tree algorithms, WEKA. .

1 INTRODUCTION

Some bad software engineer or make some faults in software as like mistake or bug. Mistaken bug is a major problem it is generally arises in coding representation and available information does not fully support to the corresponding documents.

There are many risk factors in physical word of software industry. Software industry has major need of quality, cost and time for customer satisfaction. Data mining provide a helpful way and relationship between data object to detect mistaken bug.

1.1 Classification

Classification consists of predicting a certain outcome based on a given input. In order to predict the outcome, the algorithm processes a training set containing a set of attributes and the respective outcome, usually called goal or prediction attribute. The algorithm tries to discover relationships between the attributes that would make it possible to predict the outcome. Next the algorithm is given a data set not seen before, called prediction set, which contains the same set of attributes, except for the prediction attribute – not yet known. The algorithm analyses the input and produces a prediction. The prediction accuracy defines how "good" the algorithm is. For example, in a software project data set, algorithms classify class, time fixed state, severity, priority and risk attributes for competition of software project . The training set would have relevant required information recorded previously [1].

1.2 Reduced-Error Pruning

Reduced Error Pruning As traversing over the internal nodes from the bottom to the top of a tree, the REP procedure Checks for each internal node, whether replacing it with the most repeated class that does not reduce the accuracy of trees. In this case, the node is pruned. The procedure continues until any further pruning would decrease the accuracy [2]. In order to estimate the accuracy Quinlan provides to use a pruning set. It can be shown that this procedure ends with the smallest accurate sub- tree with respect to a given pruning set [3, 4]. Example to represent REP tree with nodes and estimate the accuracy is given in Fig 1.

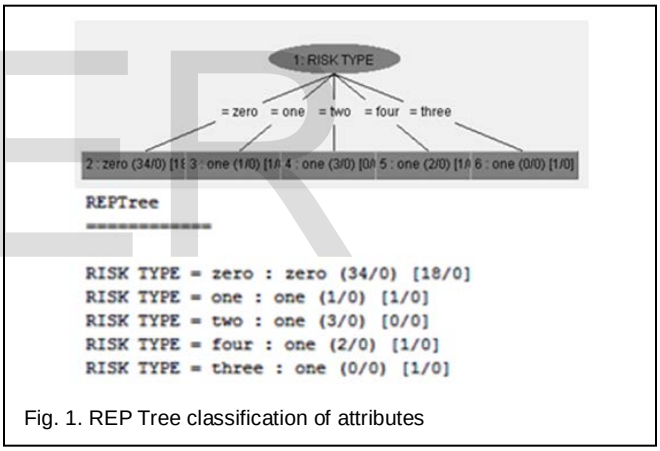
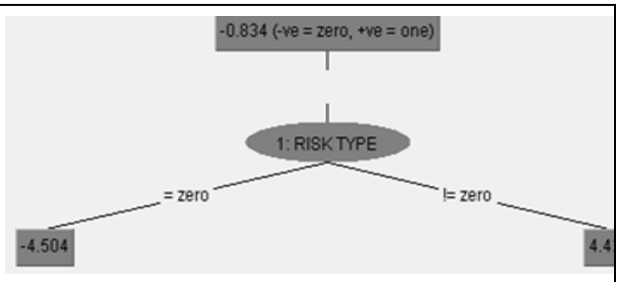


Fig. 1. REP Tree classification of attributes

1.3 AD Tree

An **alternating decision tree** (AD Tree) is a machine learning method for classification. AD Trees were introduced by you've freund and Mason. [5] However, the algorithm as presented had several type graphical errors. An alternating decision tree consists of decision nodes and prediction nodes. Decision nodes specify a predicate condition. Prediction nodes contain a single number. AD Trees always have prediction nodes as both root and leaves. An instance is classified by an AD Tree by following all paths for which all decision nodes are true and summing any prediction nodes that are traversed. For example AD tree represents decision nodes and prediction nodes as a root and leaves.

- *Dhyan Chandra Yadav is currently pursuing Ph. D. program in Computer Science from Shri Venkateshwara University, UP, Indiay, E-mail: dc9532105114@gmail.com*
- *Saurabh Pal is currently Working as Assistant Professor, MCA Dept., VBS Purvanchal University, Jaunpur, India, PH-09889807337. E-mail: drsaurabhpal@yahoo.co.in*



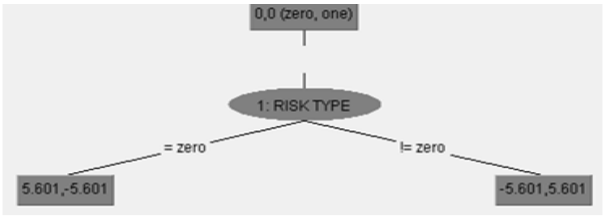

```
Alternating decision tree:

: -0.834
|   (1)RISK TYPE = zero: -4.504
|   (1)RISK TYPE != zero: 4.423
Legend: -ve = zero, +ve = one
Tree size (total number of nodes): 4
Leaves (number of predictor nodes): 3
```

Fig.2.AD Tree classification of attributes

1.4 LAD Tree

Logical Analysis of Data (LAD) tree Boros, Hammer, T. Ibaraki, Kogan, Mayoraz, and Muchnik introduced about [6] software requirements specifications are then classified into different categories like risk or no risk an each parameter in data set have his specific meaning. Logical Analysis of Data (LAD) tree is the classifier for binary target variable based on learning a logical expression that can distinguish between positive and negative samples in a data set. The central concept in LAD tree algorithm is that of classification, clustering, and other problems. The construction of LAD model for a given data set typically involves the generation of large set patterns and the selection of a subset of them that satisfies the above assumption such that each pattern in the model satisfies certain requirements in terms of prevalence and homogeneity. For example LAD tree represents risk or loss in project.




```
weka.classifiers.trees.LADTree:

: 0,0
|   (1)RISK TYPE = zero: 5.601,-5.601
|   (1)RISK TYPE != zero: -5.601,5.601
Legend: zero, one
#Tree size (total): 4
#Tree size (number of predictor nodes): 3
#Leaves (number of predictor nodes): 2
#Expanded nodes: 28
#Processed examples: 1159
#Ratio e/n: 41.392857142857146
```

Fig.3. Lad Tree classification of attributes

In mathematics, the geometric mean is a type of mean or average, which indicates the central tendency or typical value of a set of numbers by using the product of their values (as opposed to the arithmetic mean which uses their sum). The geometric mean is defined as the nth root of the product of n numbers, i.e., for a set of numbers $\{X_i\}_i{}^n$ =1,the geometric mean is defined as

$$\left(\prod_{i=1}^{N} x_i\right)^{1/N}$$

Youden's J statistic (also called Youden's index) is a single statistic that captures the performance of a diagnostic test. J = Sensitivity + Specificity – 1 with the two right-hand quantities being sensitivity and specificity.

2 Related Work

Shepperd, Schofield and Kitchenham [7] discussed that need of cost estimation for management and software development organizations and give the idea of prediction also give the methods for estimation.

Pal and Pal [8] conducted study on the student performance based by selecting 200 students from BCA course. By means of ID3, c4.5 and Bagging they find that SSG, HSG, Focc, Fqual and FAIn were highly correlated with the student academic performance.

Alsmadi and Magel [9] discussed that how data mining provide facility in new software project its quality, cost and complexity also build a channel between data mining and software engineering.

Yadav and Pal [10] use the ID3 decision tree to generate the important rules that can help to predict student enrollment into an academic programme called the Master of Computer Application. The generated tree yields that Bachelor of Science students in mathematics and computer applications will enroll and will likely to perform better as compared to Bachelor of Science students without any background in mathematics.

Boehm, Clark, Horowitz, Madachy, Shelby and Westland [11] discussed that some software companies suffer from some accuracy problems depend on his data set after prediction software company provide new idea to specify project cost schedule and determine staff time table.

K.Ribu [12] discussed that the need of open source code projects analyzed by prediction and get estimating object oriented software project by case model.

Nagwani and Verma [13] discussed that the prediction of software defect (bug) and duration similar bug and bug average in all software summery, by data mining also discuss about software bug.

Yadav and Pal [14, 15] discussed the use of different classification algorithms using standard quality of software data sets and compared the accuracy level of each method.

Hassan [16] discussed that the complex data source(audio, video, text etc.) need more of buffer for processing it does not support general size and length of buffer.

Li and Reformate [17] discussed that .the software configuration management a system includes documents, software code, status accounting, design model defect tracking and also include revision data.

Elcan [18] discussed that COCOMO model pruned accurate cost estimation and there are many thing about cost estimation because in project development involve more variable so COCOMO measure in term effort and metrics.

Chang and Chu [19] discussed that for discovering pattern of large database and its variables also relation between them by

association rule of data mining.
Kotsiantis and Kanellopoulos [20] discussed that high severity defect in software project development and also discussed the pattern provide facility in prediction and associative rule reducing number of pass in database.
Chaurasia and Pal [21, 22] conducted study on the prediction of heart attack risk levels from the heart disease database with data mining technique like Naïve Bayes, J48 decision tree and Bagging approaches and CART, ID3 and Decision Table. The outcome shows that bagging techniques performance is more accurate than Bayesian classification and J48.
Pannurat, N.Kerdprasop and K.Kerdprasop [23] discussed that association rule provide facility the relationship among large dataset as like software project term hug amount , cost record and helpful in process of project development.
Fayyad, Piatesky Shapiro, Smuth and Uthurusamy [24] discussed that classification creates a relationship or map between data item and predefined classes.
Shtern and Vassillios [25] discussed that in clustering analysis the similar object placed in the same cluster also sorting attribute into group so that the variation between clusters is maximized relative to variation within clusters.
Runeson and Nyholm [26] discussed that code duplication is a problem which is language independent. It is appear again and again another problem report in software development and duplication arises using neural language with data mining.
Vishal and Gurpreet [27] discussed that data mining analyzing information and research of hidden information from the text in software project development.
Lovedeep and Arti [28]data mining provide a specific platform for software engineering in which many task run easily with best quality and reduce the cost and high profile problems.
Nayak and Qiu [29] discussed that generally time and cost, related problems arises in software project development these problems mentation in problem report ,data mining provide help in to reduce problems also classify and reduce another software related bugs .

The proposed system will analyze risk of software defects predicts. Predicts categorical class level classifiers based on training set and the values in the class level attribute use the model in classifying new data. We compare between AD Tree, RFP Tree and LAD Tree for probability of detection, probability of false alarm, geometric mean, J static coefficient parameter (for sensitivity and specificity),correctness, completeness, average absolute error and average relative error.

3 METHODOLOGY

Our research approach is to use AD Tree, RFP Tree and LAD Tree; to model the relationships between the measurable properties of a software product and its quality. The research methodology is divided into 5 steps to achieve the desired results:
Step 1: In this step, prepare the data and specify the source of data.
Step 2: In this step select the specific data and transform it into different format by weka.
Step 3: In this step, implement data mining algorithms and checking of all the relevant bugs and errors is perform.
***Step 4*:** We classify the relevant bugs using AD Tree, RFP Tree and LAD Tree algorithm at particular time.
***Step 5*:** At the end, the results are display and evaluated completed,.

3.1 Data Preparation

TABLE 1
THE SOFTWARE BUG DEPENDABLE VARIABLES USED IN THE COMPUTATIONAL TECHNIQUE

PROPERTY	DESCRIPTION
SOURCE	Name of a project or department in MASC that raises the PR.
MEASUREMENT TYPE	(Duplicate-Bug)Srs With Metrics Count
SAMPLE SIZE	61 TOTAL:9 SRS not specify means BUG arises and 52 SRS specify means NONBUG arises in software bug-tracking system,
DEPENDABLE VARIABLE	Description
SRS(0)	0= Software Requirement Specify.
SRS (1)	1=Software Requirement not fully Specify.

An error in problem report can be corrected. All problem reports can be grouped in two categories: risk and non-risk. In risk an error can be occur automatically in software.
A software defect tracking system, GANTS which is a bug tracking system in software bug .It is set on MASC intranet to collect and maintain all problem reports from every department of MASC. The mistaken-bug is a part of class field in code implementation. Now performing for classification of

TABLE 2
THE SOFTWARE BUG EXPLANATORY VARIABLES USED IN THE COMPUTATIONAL TECHNIQUE

Explanatory Variable	**Description**
Severity	{1=normal,0=serious}describe the severity of problem report
Class	{0=sw-bug, 1=doc bug,2=change request,3=support,4=mistaken,5=duplicate}category of bug class
State	{0=closed,1=open,2=active,3=analysed,4=suspended,5=resolved,6=feedback}status of problem report analysis/non analysis
Time To Fix	{0=withintwodays, 1=within one week,2=within two week,3=within three week,4=within four week,5=within five week}take time duration in of problem report
Priority	{0=not,1=high,2=medium,3=low}describe schedule permit duration
Risk Type	{0=not,1=high,2=midium,3=low,4=cosmetic}risks can be defined as uncertainty and loss in project process.

mistaken-bug using several standard data mining tasks, data preprocessing, clustering, classification, association and tasks are needed to be done. The database is designed in MS-Excel, MS word 2010 database and database management system to store the collect data. The data is formed according to the required

Table.2. represents explanatory variables severity, state, time to fixed, priority and risk type. All variables have his specific meaning and corresponding values.

3.2 Data Selection And Transformation

In this step only those fields were selected which were required for data mining. A few derived variables were selected. Where some of the information for the variables was extracted from the database. All the predictor and response variables which were derived from database are given in table- 1 and table- 2 for reference. The survey uses status of problem report analysis /non analysis and the operationalization of the survey for items is as follows:

0=NOT,

1=HIGH,

2=MEDIUM,

3=LOW,

4=COSMETIC

The domain values for some of the variables were defined in the Table 1, 2. Working of algorithm using data mining tool software risk management contains large dataset, as the population increases day by day. Format and structures of data is converted to ARFF (attribute relation file format) format to process in weka. An ARFF file is an ASCII text file that describes a list of instances sharing a set of attributes.

```
Legend: -ve = zero, +ve = one
Tree size (total number of nodes): 4
Leaves (number of predictor nodes): 3

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          61               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0.0089
Root mean squared error                  0.0125
Relative absolute error                  3.4224 %
Root relative squared error              3.5148 %
Total Number of Instances               61

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        zero
                 1         0          1         1         1          1        one
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

  a  b   <-- classified as
 52  0 |  a = zero
  0  9 |  b = one
```

Fig.4. Instances Classified by AD Tree

```
#Expanded nodes: 28
#Processed examples: 1159
#Ratio e/n: 41.392857142857146

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          61               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0.0052 %
Root relative squared error              0.0038 %
Total Number of Instances               61

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        zero
                 1         0          1         1         1          1        one
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

  a  b   <-- classified as
 52  0 |  a = zero
  0  9 |  b = one
```

Fig.5. Instances Classified by Lad Tree

```
Size of the tree : 6

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          60               98.3607 %
Incorrectly Classified Instances         1                1.6393 %
Kappa statistic                          0.9317
Mean absolute error                      0.014
Root mean squared error                  0.1094
Relative absolute error                  5.3715 %
Root relative squared error             30.7669 %
Total Number of Instances               61

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.111     0.981     1         0.99       1        zero
                 0.889     0         1         0.889     0.941      1        one
Weighted Avg.    0.984     0.095     0.984     0.984     0.983      1

=== Confusion Matrix ===

  a  b   <-- classified as
 52  0 |  a = zero
  1  8 |  b = one
```

Fig.6. Instances Classified by RFP Tree

3.3 Data Mining Implementation

Weka is open source software that implements a large collection of machine learning algorithms and is widely used in data mining applications. From the above data bug.arff file was created. This file was loaded into weka explorer and analyzes risk of software defects predicts. Predicts categorical class level classifiers based on training set and the values in the class level attribute use the model in classifying new data.

TABLE 3
CONFUSION MATRIX FOR PREDICTION

	NO (Prediction)	YES (Prediction)
NO (Actual)	True Negative (TN) A	False Positive (FP) B
YES (Actual)	False Negative (FN) C	True Positive (TP) D

The algorithms performance is partitioned into several sub items for easier analysis and evaluation. In first part FPR, FNR G-Mean, F-Measure, J-Coefficient, Specificity and Sensitivity values are used in tabular form and Error, J-coefficient, Correctness and Completeness analyzed as a graphical format. All measures can be calculated based on four values, namely True Positive (TP, a number of correctly classified that an instances positive), False Positive (FP, a number of incorrectly classified that an instance is positive), False Negative (FN, a number of incorrectly classified that an instance is negative), and True Negative (TN, a number of correctly classified that an instance is negative). These values are defined in Table 4, 5 and Fig.7.

In second part we analyses and measure the performance from Table.5 and Fig.8 of MAE, RMSE, RAE, RRSE and time taken in build model and finally conclude the best tree algorithms for documentary requirement data set.

TABLE 4
REPRESENTS ERROR PARAMETERS BY DIFFERENT ALGORITHMS

Algorithms	AD	LAD	REP
FPR	0	0	0
FNR	0	0	0.11
G-Mean (1)	1	1	0.93
G-Mean (2)	1	1	0.93
F-Measure	1	1	0.94
J-Coefficient	1	1	0.99
Specificity	1	1	1
Sensitivity	1	1	0.99

3.4 Result And Discussion

There are several algorithms for classification of which the most well-known and widely applicable ones are run on the given dataset. The results of each of these runs using weka are provided below.

Experiment-A

Given Table 4 shows the performance between the AD Tree, Lad Tree And REP Tree algorithms for FPR, FNR G-Mean, F-Measure, J-Coefficient, Specificity and Sensitivity values. It is clear that in the given Table 4 AD Tree and LAD Tree perform the same response but REP give the difference and less performance.

The false positive rate (FPR), the false negative rate (FNR), and error parameters are used for performance evaluation.

FPR=FP/(FP+TN) -(1)
FNR=FN/(FN+TP) -(2)
Error=(FN=FP)/TP+FP+FN+TN) -(3)

These performance indicators should be minimized, but there is a trade-off between the FPR and FNR values. Correctness and completeness parameters were used for the evaluation of fault prediction models. Formulas 15 and 16 show how to calculate correctness and completeness measures.

Correctness =TP/(FP+TP) -(4)
Completeness =TP/(FN+TP) -(5)

Proposed the usage of the J parameter to measure the accuracy of binary classifiers in software engineering. The J coefficient is calculated by using sensitivity and specificity parameters.

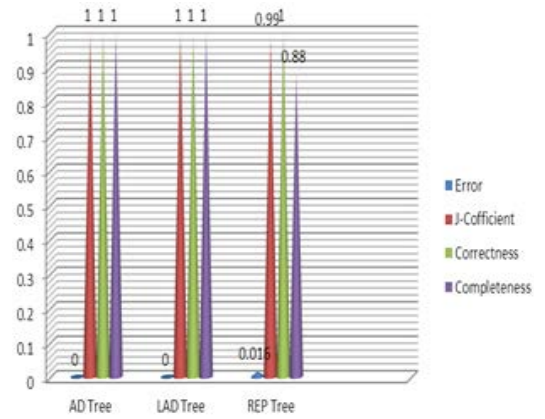


Fig.7.Graph represents comparison of Error, Sensitivity, specificity, and the J parameter

Sensitivity, specificity, and the J parameter are calculated by using Formulas 9, 10, and 11 respectively.

Sensitivity =TP/(TP+FN) -(6)
Specificity =TN/(TN+FP) -(7)
J = sensitivity + specificity – 1 -(8)

Sensitivity measures the ratio of actual faulty modules which are correctly identified and specificity measures the ratio of non-faulty modules which are correctly identified. Some researchers use G-mean1, G-mean2, and F-measure metrics for the evaluation of prediction systems, which are built on imbalanced datasets. Formulas,, and show how to calculate these measures, respectively. Formula is used for precision parameter and True Negative Rate (TNR) is calculated by using Formula.

True Negative Rate (TNR) =TN/(TN+FP) (9)
G-mean1= $\sqrt{(\text{Precision} * recall)}$ (10)
G-mean2 = $\sqrt{recall * TNR}$ (11)
F-measure =2(recall*precision)/ (recall+precision) (12)

Used G-mean1, G-mean2, and G-mean3 to benchmark several machine learning algorithms for software fault prediction. They sorted algorithms according to their performance results for each metric and marked the top three algorithms for each metric. They identified the algorithm that provides G-mean1, G-mean2, and F-measure values in the top three. According to this study, Balanced Random Forests is the best algorithm for software fault prediction problems.

Evaluated the performance of classifiers according to the F-measure value.

Accuracy=(TN+TP)/(TN+FP+FN+TP) (13)

The following graph shows the comparison of AD Tree, LAD Tree and REP Tree algorithms with the corresponding values of Error, J-coefficient, correction and completeness.

- AD Tree and LAD Tree performed the same result for the attribute classification values of Error, J-coefficient, Correctness and Completeness.
- REP Tree algorithm have poor performance for attribute value of Error, J-coefficient, Correctness and Completeness

So we discussed the result in the Experiment B.

Experiment –B

All measures can be calculated based on four values, namely True Positive (TP, a number of correctly classified that an instances positive), False Positive (FP, a number of incorrectly classified that an instance is positive), False Negative (FN, a number of incorrectly classified that an instance is negative), and True Negative (TN, a number of correctly classified that an instance is negative). These values are defined in Table 4, 5

Fig.8.Graph represents comparison of Time, RRSE, RAE, RMSE, MAE and Accuracy.

Given table and graph shows the comparison between the classified attribute by weka tool.
From table and graph it is clear-

- LAD Tree algorithm give less error value of MAE compare to AD and REP algorithms.
- LAD Tree algorithms give less error value of RMSE compare to AD and REP algorithms.
- LAD Tree algorithms give less error value of RAE compare to AD and REP algorithms.
- LAD Tree algorithms give less error values of RRSE compare to AD and REP algorithms.

Only REP takes some less time compare to AD and LAD Tree algorithms. The time difference is minor so it is clear Lad Tree give best result compare to AD and REP in weka.

As demonstrated in this document, the numbering for sections upper case Arabic numerals, then upper case Arabic numerals, separated by periods. Initial paragraphs after the section title are not indented. Only the initial, introductory paragraph has a drop cap.

TABLE 5
ERROR COMPARISON BETWEEN ALGORITHMS

Tree Algorithm	AD	LAD	REP
MAE	0.008	0	0.014
RMSE	0.01	0	0.109
RAE	3.42	0.005	5.37
PRSE	3.51	0.003	30.76
Accuracy	1	1	3.5
Time (Sec)	0.03	0.03	0.01

4 CONCLUSION

In this paper were done experiments with Weka Machine Learning Tool in order to choose the best Data Mining tree algorithms to be applied over selected datasets. Incompleteness in requirements documents is also checked. Quality factors such as completeness, correctness, understanding etc. are discussed and their corresponding formulae are mentioned. These requirements metrics help in identifying and rectifying errors in requirements document. However, research in this area is in continual progress to provide better metrics for Product, People and Process to support the development of software. Implementation of quality metrics during the development process ensures production of high quality software. In this paper three different classifiers and results are evaluated from the Experiment-A and Experiment-B based on the proposed statistical methods. The results confirm that for LAD Tree algorithms is a best classifier in comparison of AD Tree and REP Tree algorithms. So the future work will be based on other classifiers that can be applied on the data set and also to apply other data mining tools on the data set such that the best techniques can be identified.

REFERENCES

[1] Jiawei Han, Micheline Kamber and Jian Pei "Data Mining Concepts and Techniques" Morgan Kaufmann Publishers Third Edition. 2012 .

[2] A. S. Galathiya, A. P. Ganatra and C. K. Bhensdadia," : ImprovedDecision Tree Induction Algorithm with Feature Selection, CrossValidation, Model Complexity and Reduced Error Pruning", A. S.Galathiya et al, / (IJCSIT) International Journal of Computer Scienceand Information Technologies, Vol. 3 (2) , 2012,3427-3431.

[3] Tapio Elomaa ,Matti Kääriäinen," An Analysis of Redued Error Pruning", Department Journal of Artificial Intelligence Research 15 (2001)163-187.

[4] Lior Rokach and Oded Maimon, "Top-Down Induction of DecisionTrees Classifiers—A Survey", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 35, NO. 4, NOVEMBER 2005.

[5] Yoav Freund and Llew Mason. The Alternating Decision Tree Algorithm. Proceedings of the 16th International Conference on Machine Learning, pages 124-133 (1999).

[6] E. Boros, P. L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik, "An implementation of logical analysis of data," *IEEE Trans. On Knowl. Data Eng*, vol. 12, no. 2, pp. 292–306, 2000.

[7] M. Shepperd, C. Schofield, and B. Kitchenham, "Effort estimation using analogy," in of the 18th International Conference on Software Engineering,pp.170-178. Berlin, Germany, 1996.

[8] Pal, A. K., & Pal, S., "Classification model of prediction for placement of students", International Journal of Modern Education and Computer Science (IJMECS), 5(11),2013, 49.

[9] Alsmadi and Magel, "Open source evolution Analysis," in proceeding of the 22nd IEEE International Conference on Software Maintenance (ICMS'06),

phladelphia, pa.USA, 2006.

[10] Yadav, Surjeet Kumar, and Saurabh Pal. "Data mining application in enrollment management: A case study." International Journal of Computer Applications (IJCA) 41.5, 2012, 1-6.

[11] Boehm, Clark, Horowitz, Madachy, Shelbyand Westland, "Cost models for future software lifecy cle Process COCOMO2.0."in Annals of software Engineering special volume on software process and prodocuct measurement, J.D.Arther and S.M.Henry, Eds, vol.1, pp.45-60, j.c. Baltzer AG,science publishers, Amsterdam,The Netherlnds, 1995.

[12] Ribu,Estimating object oriented software projects With use cases,M.S.thesis, University of Oslo Department of informatics, 2001.

[13] N. Nagwani and S. Verma,"prediction data mining Model for software bug estimation using average Weighted similiarity,"In proceeding of advance Computing conference (IACC), 2010.

[14] Yadav, Dhyan Chandra, and Saurabh Pal. "Analysis Receiver Operating Characteristics of Software Quality Requirement by Classification Algorithms Analysis" IJCA 116.8, 2015.

[15] Yadav, Dhyan Chandra, and Saurabh Pal. "Software Bug Detection using Data Mining." IJCA, 115.15, 2015

[16] A.E.Hassan ,"The road ahead for mining software Repositories", in processing of the future of software Maintenance at the 24th IEEE international Conference on software maintenance, 2008.

[17] Z.Li and Reformat,"A practical method for the Software fault prediction",in proceeding of IEEE Nation conference information reuse and Integration (IRI), 2007.

[18] C.Elcan,"The foundations of cost sensitive learning In processing of the 17 International conference on Machine learning, 2001.

[19] C.Chang and C.Chu,"software defect prediction Using international association rule mining", 2009.

[20] S.Kotsiantis and D.Kanellopoulos,"Associan rule mining:Arecent overview",GESTS international transactiona on computer science science and Engineering, 2006.

[21] Chauraisa V. and Pal S., "Data Mining Approach to Detect Heart Diseases", International Journal of Advanced Computer Science and Information Technology (IJACSIT),Vol. 2, No. 4,2013, pp 56-66.

[22] Chauraisa V. and Pal S., "Early Prediction of Heart Diseases Using Data Mining Techniques", Carib.j.SciTech,,Vol.1, pp. 208-217, 2013.

[23] N.Pannurat,N.Kerdprasop and K.Kerdprasop"Database reverses engineering based On Association rule mining",IJCSI international Journal Of computer science issues 2010.

[24] U.M.Fayyad,G Piatesky Shapiro,P.Smuth and R.Uthurusamy,"Advances in knowledge discovery And data mining",AAAI Press,1996.

[25] M.Shtern and Vassilios,"Review article advances in Software engineering clustering methodologies for Software engineering",Tzerpos volume,2012.

[26] P.Runeson and O.Nyholm,"Detection of duplicate Defect report using neural network processing",in Proceeding of the 29th international conference on Software engineering 2007.

[27] G.Vishal and S.L. Gurpreet,"A servey of text mining Techniques and applications",journal of engineering Technologies in web intelligence, 2009.

[28] Lovedeep and VarinderKaurArti" Application of Data mining techniques in software engineering"International journal of electrical,electronics and computer system(IJEECS) Volume-2 issue-5, 6. 2014.

[29] Richi Nayak andTianQiu"Adata mining application"international journal of software engineeringandKnowledgeengineering volume.15, issue-04,2005.