ISSN 2229-5518

Analysis of Software Defect Classes by Data Mining Classifier Algorithms

Dhyanchandra Yadav, Rajeev Kumar

Abstract— Software bugs create problems in software project development. We can categories software bugs by some specific data mining classifiers algorithms. 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, Random Tree,ID3,J48,J48 graft, OneR, ZeroR, Prism ,Bayes Net and Naïve Bayes for correctly classify and uncorrectly classify with time build model.

Key Words— AD Tree, RFP Tree and LAD Tree, Random Tree, ID3, J48, J48 graft, OneR, ZeroR, Prism, Bayes Net , Naïve Bayes and Weka Tool.

1 Introduction

Classification is the process of organizing data into categories for its most effective and efficient use. Classification is the process of predicting the class of a new item. Therefore to classify the new item and identify to which class it belongs given a collection of records (*training set*). Each record contains a set of *attributes*, one of the attributes is the *class*. In our hole analysis we use classifiers algorithms as[1]:

1.1 Classifiers Bayes

Bayes Net and Naïve Bayes classifiers provide help in software bug and non_bug classification. In classification the Bayes classifier minimises the probability of misclassification, or risk, of a classifier [2].

1.2 Classifiers Rules

OneR, ZeroR, Prism provide help in data classification as zer leve ,one level and calculate probability of incidences. Classification rule is a procedure in which Bayes have event and supporting evidence by which arises many cases[3][4][5].

1.3 Classifiers Tree

Decision tree classification algorithm is widely used in statistics, data mining, machine learning. The goal is to create a decision tree model, which uses the given input data to predict the target data classification. For the nodes within the tree, we compare the attribute values. Each branch is a possible classification for the target data. Leaf node is the classification of the target data. Decision tree is a classifier of root node which generate another branches as a node[2].

- *Dhyan Chandra Yadav is research scholar in Ph.D program in Computer SVU,Gajraula,Amroha (U.P.)E-mail:dc9532105114@gmail.com .*
- *Rajeev Kumar is currently working as Assistant Professor, SVU,Gajraula,Amroha (U.P.) India. Dept, of Computer Science., PH-09997672215. E-mail: rajeev2009mca@gmail.com*

2 Related Work-

Shepperd, Schofield and Kitchenham[6] discussed that need of cost estimation for management and software development organizations and give the idea of prediction also give the methods for estimation.

Alsmadi and Magel[7] 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.

Boehm,Clark,Horowitz,Madachy,Shelby and Westland[8] 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[9] 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[10] 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.

Hassan [11] discussed that the complex data source (audio, video, text etc.) need more have buffer for processing it does not support general size and length of buffer.

Li and Reformate[12] discussed that .the software configuration management a system includes documents, software code, status accounting, design model defect tracking and also include revision data.

Elcan[13] 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 [14] discussed that for discovering pattern of large database and its variables also relation between them by association rule of data mining.

Kotsiantis and Kanellopoulos[15] 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.

Pannurat, N.Kerdprasop and K.Kerdprasop[16] 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,PiateskyShapiro,Smuth and Uthurusamy [17] discussed that classification creates a relationship or map between data item and predefined classes.

Shtern and Vassillios[18] 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[19] 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[20] discussed that data mining analyzing information and research of hidden information from the text in software project development.

Lovedeep and Arti[21]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[22] 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 type of software defect. 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, Random Tree,ID3,J48,J48 graft, OneR, ZeroR, Prism ,Bayes Net and Naïve Bayes for correctly classify and uncorrectly classify with time build model.

3 Methodology-

Our research approach is to use some classifiers algorithms (trees, rules and Bayes). 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 classifier algorithms at particular time.

Step 5: At the end, the results are display and evaluated completed,.

3.1 Data Preparation-

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".

TABLE 1
THE VARIABLES USED IN THE COMPUTATIONAL TECHNIQUE

DEPENDABLE VARIABLE	DETAILS
{NON-BUG=0}	No loss in project development process.
{SOFT-BUG=1}	Software defect in project development process.
{DOC-BUG=2}	Software defect in project development process.
{MISTAKEN- BUG=3}	Software defect in project development process.
{DUPLICATE-BUG=4}	Software defect in project development process.
EXPLNATORY VARIABLE	VALUE
SEVERITY	{1=Normal,0=Serious}
NOT REDUNDANT	1= No Redundancy. 0= Complete Redundant.
STATE	{0=Closed,1=Open,2=Active,3=Analysed,4=Suspended,5=Resolved,6=Feedback}
TIME TO FIX	{0=Within Two Days,1=Within One Week,2=Within Two Week,3=Within Three Week,4=Within Four Week,5=Within Five Week}
PRIORITY	{0=Not,1=High,2=Medium,3=Low}
RISK TYPE	{0=Not,1=High,2=Midium,3=Low,4=Cosmetic}

The soft-bug, doc-bug, mistaken-bug and duplicate-bug are parts of class field in software development. Now performing for classification of software defect using several standard algorithms of data mining classifier algorithms. The database is designed in "MS-Excel, MS word 2010 database" and database management system to store the collect data.

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. We select some classifier algorithms and transform all classifiers in specific way as:

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.csv 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.

ISSN 2229-5518

Time taken to build model: 0.02 seconds

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

Correctly Classified Instances	49	81.6667 %
Incorrectly Classified Instances	7	11.6667 %
Kappa statistic	0.2768	
Mean absolute error	0.0798	
Root mean squared error	0.2208	
Relative absolute error	64.8612 %	
Root relative squared error	99.603 %	
UnClassified Instances	4	6.6667 %
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.75	0.889	1	0.941	0.694	zero
	1	0.018	0.5	1	0.667	0.741	four
	0	0	0	0	0	0.441	three
	0	0	0	0	0	0.5	two
	0	0	0	0	0	0.5	one
Weighted Avg.	0.875	0.643	0.771	0.875	0.819	0.663	

Fig. 1. Instances classified by Id3 algorithms.

Time taken to build model: 0 seconds

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

Correctly Classified Instances	50	83.3333 %
Incorrectly Classified Instances	10	16.6667 %
Kappa statistic	0.1513	
Mean absolute error	0.1028	
Root mean squared error	0.2445	
Relative absolute error	71.3304 %	
Root relative squared error	95.8973 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.909	0.831	1	0.907	0.609	zero
	0.5	0	1	0.5	0.667	0.953	four
	0	0	0	0	0	0.33	three
	0	0	0	0	0	0.432	two
	0	0	0	0	0	0.441	one
Weighted Avg.	0.833	0.742	0.712	0.833	0.763	0.582	

Fig. 2. Instances classified byJ48 algorithms.

Time taken to build model: 0 seconds

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

Correctly Classified Instances	50	83.3333 %
Incorrectly Classified Instances	10	16.6667 %
Kappa statistic	0.1513	
Mean absolute error	0.1028	
Root mean squared error	0.2445	
Relative absolute error	71.3304 %	
Root relative squared error	95.8973 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.909	0.831	1	0.907	0.609	zero
	0.5	0	1	0.5	0.667	0.953	four
	0	0	0	0	0	0.33	three
	0	0	0	0	0	0.432	two
	0	0	0	0	0	0.441	one
Weighted Avg.	0.833	0.742	0.712	0.833	0.763	0.582	

Fig. 3. Instances classified byJ48 graft algorithms

Time taken to build model: 0.03 seconds

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

Correctly Classified Instances	50	83.3333 %
Incorrectly Classified Instances	10	16.6667 %
Kappa statistic	0.3576	
Mean absolute error	0.0939	
Root mean squared error	0.2427	
Relative absolute error	65.1274 %	
Root relative squared error	95.1993 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.98	0.545	0.889	0.98	0.932	0.641	zero
	1	0	1	1	1	1	four
	0	0.038	0	0	0	0.445	three
	0	0	0	0	0	0.915	two
	0	0.034	0	0	0	0.042	one
Weighted Avg.	0.833	0.45	0.759	0.833	0.794	0.625	

Fig. 4. Instances classified by Lad tree algorithms.

Time taken to build model: 0 seconds

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

Correctly Classified Instances	50	83.3333 %
Incorrectly Classified Instances	10	16.6667 %
Kappa statistic	0.2042	
Mean absolute error	0.0956	
Root mean squared error	0.2498	
Relative absolute error	66.3074 %	
Root relative squared error	97.9673 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0.818	0.845	1	0.916	0.646	zero
	0.5	0.017	0.5	0.5	0.5	0.961	four
	0	0	0	0	0	0.381	three
	0	0	0	0	0	0.475	two
	0	0	0	0	0	0.483	one
Weighted Avg.	0.833	0.669	0.707	0.833	0.765	0.62	

Fig. 5. Instances classified by Random Tree algorithms.

Time taken to build model: 0 seconds

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

Correctly Classified Instances	49	81.6667 %
Incorrectly Classified Instances	11	18.3333 %
Kappa statistic	0	
Mean absolute error	0.1279	
Root mean squared error	0.2538	
Relative absolute error	88.7089 %	
Root relative squared error	99.5231 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	1	0.817	1	0.899	0.45	zero
	0	0	0	0	0	0.086	four
	0	0	0	0	0	0.33	three
	0	0	0	0	0	0.042	two
	0	0	0	0	0	0.042	one
Weighted Avg.	0.817	0.817	0.667	0.817	0.734	0.41	

Fig. 6 Instances classified by REP Tree algorithms.

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances          49               81.6667 %
Incorrectly Classified Instances        11               18.3333 %
Kappa statistic                          0
Mean absolute error                      0.1442
Root mean squared error                  0.255
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1         0.817     1         0.899      0.45     zero
                 0         0         0         0         0          0.086    four
                 0         0         0         0         0          0.33     three
                 0         0         0         0         0          0.042    two
                 0         0         0         0         0          0.042    one
Weighted Avg.    0.817     0.817     0.667     0.817     0.734      0.41
```

Fig. 7. Instances classified by ZeroR algorithms .

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances          50               83.3333 %
Incorrectly Classified Instances        10               16.6667 %
Kappa statistic                          0.2042
Mean absolute error                      0.0667
Root mean squared error                  0.2582
Relative absolute error                 46.2382 %
Root relative squared error            101.2676 %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.818     0.845     1         0.916      0.591    zero
                 0.5       0.017     0.5       0.5       0.5        0.741    four
                 0         0         0         0         0          0.5      three
                 0         0         0         0         0          0.5      two
                 0         0         0         0         0          0.5      one
Weighted Avg.    0.833     0.669     0.707     0.833     0.765      0.582
```

Fig. 8. Instances classified by OneR algorithms.

```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances          49               81.6667 %
Incorrectly Classified Instances         8               13.3333 %
Kappa statistic                          0.3028
Mean absolute error                      0.0561
Root mean squared error                  0.2369
Relative absolute error                 43.3167 %
Root relative squared error            101.5941 %
UnClassified Instances                   3                5      %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.667     0.889     1         0.941      0.717    zero
                 1         0.036     0.333     1         0.5        0.733    four
                 0         0         0         0         0          0.5      three
                 0         0         0         0         0          0.5      two
                 0         0         0         0         0          0.5      one
Weighted Avg.    0.86      0.562     0.754     0.86      0.801      0.687
```

Fig. 9. Instances classified by Prism algorithms.

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances          48               80      %
Incorrectly Classified Instances        12               20      %
Kappa statistic                          0.2996
Mean absolute error                      0.0825
Root mean squared error                  0.2475
Relative absolute error                 57.2014 %
Root relative squared error             97.0712 %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.939     0.545     0.885     0.939     0.911      0.642    zero
                 1         0.034     0.5       1         0.667      1        four
                 0         0.038     0         0         0          0.342    three
                 0         0.034     0         0         0          0.915    two
                 0         0         0         0         0          0.949    one
Weighted Avg.    0.8       0.452     0.739     0.8       0.766      0.629
```

Fig. 10. Instances classified by Bayes Net algorithms.

```
Time taken to build model: 0 seconds

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

Correctly Classified Instances          49               81.6667 %
Incorrectly Classified Instances        11               18.3333 %
Kappa statistic                          0.3299
Mean absolute error                      0.0797
Root mean squared error                  0.2423
Relative absolute error                 55.2623 %
Root relative squared error             95.0328 %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.959     0.545     0.887     0.959     0.922      0.635    zero
                 1         0.052     0.4       1         0.571      1        four
                 0         0.019     0         0         0          0.288    three
                 0         0.017     0         0         0          0.915    two
                 0         0         0         0         0          0.949    one
Weighted Avg.    0.817     0.45      0.738     0.817     0.772      0.616
```

Fig. 1. Instances classified by Naive Bayes algorithms.

The algorithms performance is partitioned into several sub items for easier analysis and evaluation. In first part evaluate classification and non-classification instances values with time builds models are used in tabular form. 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).

Finally conclude the best tree algorithms for software defect data set.

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

ISSN 2229-5518
are provided below.

From table .2 we easily calculate the correctly and incorrectly classified total instances of data set. Given table-.2 shows the comparison between the classified attribute by weka tool.

Since the predictive performance is fully depend on accuracy and kappa static is a metric that compute an observe accuracy with expected accuracy. If number of classes is more than two then according table.2 we easily analyzed as:

TABLE 2
COMPUTING THE CLASSIFIER ALGORITHMS ON BASIS OF MORE THAN TWO CLASSES

S.N	Classifier Algorithms	Correctly Classified	Incorrectly Classified	Time build a Model	Number of Classes	Unclassified	Kappa Static
1	ID3	81.667	11.667	0.02	5	6.667	0.2276
2	J48	83.333	16.667	0.00	5	0.000	0.1513
3	J48graft	83.333	16.667	0.00	5	0.000	0.1513
4	LAD	83.333	16.667	0.03	5	0.000	0.3576
5	Random	83.333	16.667	0.00	5	0.000	0.2042
6	REP	81.667	18.335	0.00	5	0.000	0.1279
7	ZeroR	81.667	18.335	0.00	5	0.000	0.1442
8	OneR	83.333	16.667	0.00	5	0.000	0.2042
9	Prism	81.667	13.333	0.02	5	6.000	0.3028
10	BayesNet	80.000	20.000	0.00	5	0.000	0.2996
11	Naïve Bayes	81.667	18.333	0.00	5	0.000	0.3299

- In the above table give highest number of percentage for correctly classified instances value is 83.333%.
- Total highest number of classes has a common value "five" for all classifier algorithms.
- The minimum number of unclassified instances is 0.00.
- The maximum highest number accuracy is evaluated 0.3576.

4 CONCLUSION-

In this analysis we choose the LAD Tree is the best data mining classifier algorithms to be applied over selected datasets. Because LAD Tree has highest correctly value 83.333% and minimum number of unclassified instances is 0.00. Also Lad tree have highest value of metric for accuracy. Implementation of quality metrics during the development process ensures production of high quality software. In this analysis different classifier algorithms and results are evaluated. 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] Sunita Tiwari and Neha Chaudhary "Data mining And Warehousing" Dhanpati Rai and Co.(P) Ltd. First edition: 2010. Kaufmann Publishers Third Edition. 2012.

[3] Holte, R.C., 1993 Very simple classification rules Perform well on most commonly used datasets. Machine Learning Vol 11, pp 63-91.

[4] OneR:http://en.wikipedia.org/wiki/One-attribute-rule 16 April 2007.

[5] Tzung-Pei Hong and Shian Shyong, "Tseng; Two-phase PRISM Learning Algorithms", Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, IEEE International Conference, Vol. 4, pp 3895 – 3899,1997

[6] 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.

[7] Alsmadi and Magel, "Open source evolution Analysis," in proceeding of the 22nd IEEE International Conference on Software Maintenance (ICMS'06), phladelphia, pa.USA, 2006.

[8] Boehm,Clark,Horowitz,Madachy,Shelby and Westland, "Cost models for future software life cycle 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.

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

[10] 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.

[11] 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.

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

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

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

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

[16] N.Pannurat,N.Kerdprasop and K.Kerdprasop"Database reverses engineering based On Association rule mining",IJCSI international Journal Of com-

ISSN 2229-5518
puter science issues 2010.

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

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

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

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

[21] 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.

[22] Richi Nayak andTianQiu"Adata mining application "international journal of software engineering and Knowledge engineering volume.15,issue-04,2005.