Software Bug Detection using Data Mining

Dhyan Chandra Yadav
Research Scholar, Shri Venkateshwara University,
Gajraula, Amroha (U.P.)

Saurabh Pal
Head, MCA Dept.,
VBS Purvanchal University Jaunpur (U.P.)

ABSTRACT
The common software problems appear in a wide variety of applications and environments. Some software related problems arises in software project development i.e. software related problems are known as software defect in which Software bug is a major problem arises in the coding implementation .There are no satisfied result found by project development team. The software bug problems mentation in problem report and software engineer does not easily detect this software defect but by the help of data mining classification software engineers easily can classify software bug. This paper classified and detect software bug by J48, ID3 and Naïve Bayes data mining algorithms. Comparison of these algorithms to detect accuracy and time taken to build model is also presented in this paper.

General Terms
Data Mining, Classification algorithms, Software Bug, Weka Tool.

Keywords
Classification: ID3, J48 and Naïve Bayes; Software BUG; WEKA.

1. INTRODUCTION
Humpherey [1] introduced about software bug. It is a major bug in coding implementation because without correct code we do not found the correct result. Software engineering teams have bug report in which these type bugs mentioned. In the absent of perfect (required) quality of software, customer does not satisfy. The help of software tracker software engineer easily detect error as a software defect and its type. The software bug report is known as problem report but by the help of data mining easily we detect and analyzed bug.

1.1 Data Mining
Tiwari and Chaudhary [2] introduced about data mining. It is processes in computer science by the help of this process easily extract relationship and pattern from data and collect information which provides help in decision making as we want in software development field. It is easily extract information from problem reports and take decision by the help of information we can detect software defect and improve software quality.

1.2 Classification
Tiwari and Chaudhary [2] introduced about classification which divide data samples into target classes. Classification have a training set which provide a facility to have a common level of same classes of data .Some different type bugs in software project development: SW-bug, document bug, duplicate bug and mistaken bug .These bugs have common level bug classes of data object known as software defect in training set.

1.3 Decision Tree
Tiwari and Chaudhary [2] introduced about decision tree which is a classifier of root node which generate another branches as a node. The common attributes of data at class level each node have own information. For example

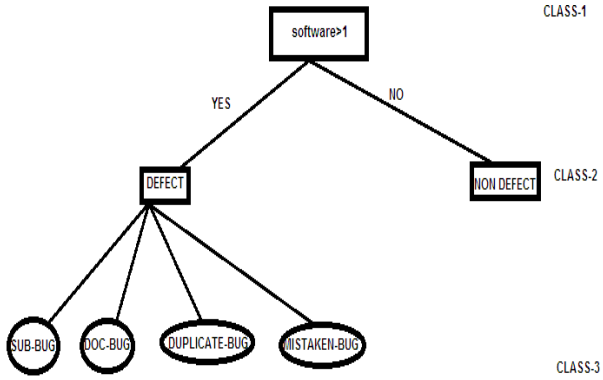

Fig 1: Represents to check level defect of software

1. **If software > 1** then root node extract another branches or internal node (not leaf node) show class (2).
2. **2-If software < 1** then shows root node on class (1).
3. **3-If software defect > 1** then found defect classification categories bug at class (3) not extract another node otherwise on the class (2).

1.4 ID3
J. Ross Quinlan [3] introduced about iterative dichotomies algorithms start with training sample of data at the root and create the partition of root which not have a common attribute with link between corresponding sub sample values and the extracted node as like the child node behave as a class on his node then all possible outcomes instances check whether they are falling under the same class or not and classifier of top to bottom and minimize the information entropy measure.

Hunt and Quinlan presents C4.5 is a successor of ID3.The C4.5 is represents by J48 in Weka. J48 classification by decision tree leaf nodes represents class level:

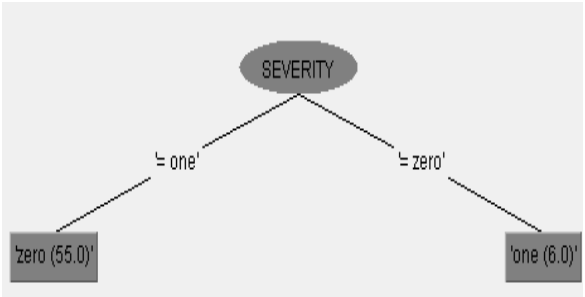

Fig 2: J48 classification of attribute by decision tree

1. A flow chart like tree structure internal node denotes a test.
2. On an attribute branch represents an out comes of the test.
3. Decision tree generate consists of two phases.

1.5 Bayes Rule

Tiwari and Chaudhary [2] introduced about Bayes rule have event and supporting evidence and there are two cases arise:

1. If event occurs means between evidence **P (H**) probability conform.
2. Event occurs means with supporting evidence **P (H/E).**

Let **H** be the event of SW-bug and **E** be the evidence of software defect then we have

P (SW-bug/software defect) =P (software defect/SW-bug)*P (SW-bug)/P (software defect)

Naïve Bayes classification algorithm basically used for high dimension input from the above example. We can predict and output of some event and observing some evidence. Generally it is better to have more than one evidence to support the prediction of an event.

2. RELATED WORK

Shepperd, Schofield and Kitchenham [4] discussed that need of cost estimation for management and software development organizations and give the idea of prediction and discuss the methods for estimation.

Alsmadi and Magel [5] 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 [6] 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.

Pal and Pal [7] 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.

K.Ribu [8] 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 [9] 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 [10] 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.

Chaurasia and Pal [11, 12] 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.

Li and Reformate [13] discussed that the software configuration management is a system includes documents, software code, status accounting, design model defect tracking and also include revision data.

Elcan [14] 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 [15] discussed that for discovering pattern of large database and its variables also relation between them by association rule of data mining.

Kotsiantis and Kanellopoulos [16] 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 [17] 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 [18] discussed that classification creates a relationship or map between data item and predefined classes.

Pal [19] conducted study on the student dropout rate by selecting 1650 students from different branches of engineering college. In their study, it was found that student's dropout rate in engineering exam, high school grade; senior secondary exam grade, family annual income and mother's occupation were highly correlated with the student academic performance.

Shtern and Vassillios [20] 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 [21] 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 [22] discussed that data mining analyzing information and research of hidden information from the text in software project development.

Lovedeep and Arti [23] 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.

Yadav and Pal [24] conducted a study using classification tree to predict student academic performance using students' gender, admission type, previous schools marks, medium of teaching, location of living, accommodation type, father's qualification, mother's qualification, father's occupation, mother's occupation, family annual income and so on. In their study, they achieved around 62.22%, 62.22% and 67.77% overall prediction accuracy using ID3, CART and C4.5 decision tree algorithms respectively.

The present study proposed classification to get better rules and to decrease the error rate as much as possible, several approaches are used SW-bug detection from a software defect origin using different data mining techniques (ID3, J48 , Naïve Bayes) and Weka tool .The aim is to detect better accurate result of data by classifying all observations of BUG.

3. METHODOLOGY

3.1 Data Preparation

Table 1. Variables used in the computational technique

PROPERTY	DESCRIPTION	
Source	Name of a project or department in MASC that raises the PR.	
Bug Type	(SW-BUG) The bug is from the software code implementation.	
Sample Size	61 TOTAL: 6 BUG and 55 NON-BUG software bug-tracking system, GNATS (A Tracking System by GNU), is set up on MASC Intranet	
Dependable Variables		
Bug (1)	BUG accepted	
Non-Bug (0)	BUG not accepted	
Property	**Value**	**Description**
Severity	{1=Normal, 0=Serious}	Describe the Severity of PR
Priority	{0=Not, 1=High, 2=Medium, 3=Low}	Describe schedule permit duration
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}	Take time duration in PR
Class	{0=SW-BUG, 1=DOC-BUG, 2=Change request, 3=Support, 4=Mistake, 5=Duplicate}	Category of BUG classes

A software error arises in problem report and all problem reports grouped in two categories: recoverable and unrecoverable. In recoverable group an error easily recovered automatically by software. A software bug tracking system GANTS, (a tracking system by GNU) is set up on MASC intranet to collect and maintain all problem reports from every department of MASC. The SW-bug is an input value for class field; SW-bug is from code implementation. Now performing for classification of SW-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 format and structures and data is converted to .csv (comma delimited) format to process in Weka that describes a list of instances sharing a set of attributes.

3.2 Data Selection and Transformation

The variables move automatic in the computational technique to identify the SW-bug or none bug in software. The kappa static is a matric that compares an observed accuracy with the expected accuracy. The kappa static is used to evaluate a single classifier and confusion matrix.

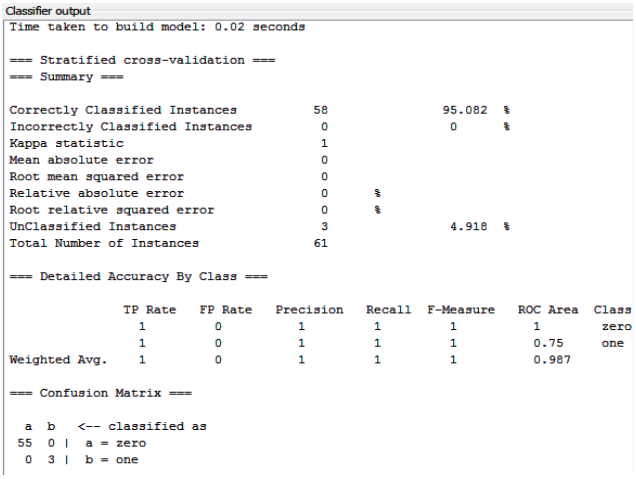

Fig 3: Instances classified by ID3 algorithm

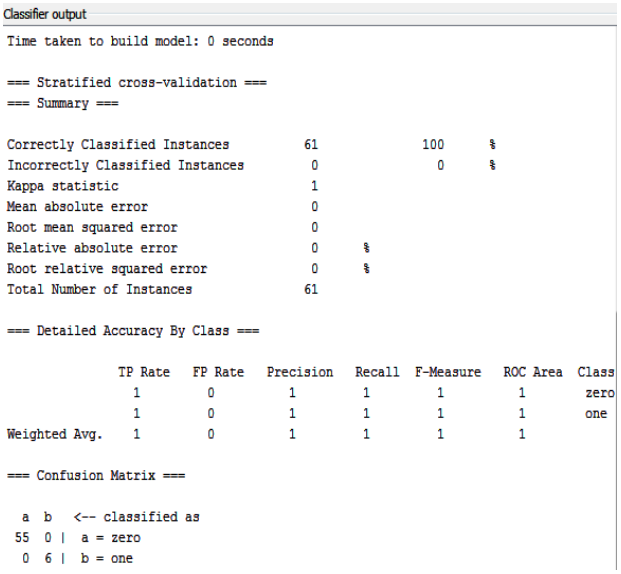

Fig 4: Instances classified by J48 algorithm

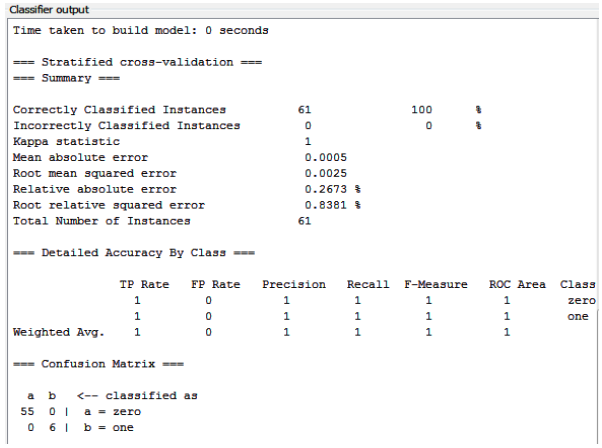

Fig 5: Instances classified by Naïve Bayes algorithm

In general, positive=identified and negative=rejected.

Therefore

True positive=correctly identified.

False positive=incorrectly identified.

True negative=correctly rejected.

False negative=incorrectly rejected.

By the help of confusion matrix easily specified layout table that allow visualization of the performance of an algorithm. Each column of matrix representation of instances is a predicted class, while each row represents the instances is an actual class.

3.3 Implementation of Data Mining

The paper presents an approach to classifying SW-bug in order to predict design, implement and evaluate a series of pattern classifier also compare performance of an online SW-bug dataset. The classifiers were used to declare surety of bug. Present paper uses the J48, Id3 and Naïve Bayes algorithms to improve the prediction accuracy. These techniques are of considerable useful in identifying software bug in very large data set.

3.4 Result and Discussion

The proposed techniques are included in Weka tool; decision tree and naïve Bayes techniques. Data mining tools are software components and proposed tool that will be applied in Weka and support several data mining task. The proposed techniques that will be applied in this paper are decision tree (J48, ID3) and Naïve Bayes because it is powerful classification algorithms.

From the table 2 it is clear that kappa static value observed give the equal value compare to J48 and NB algorithms and ID3. Naive Bayes give the more error compare to J48 and ID3 algorithms. But ID3 and J48 take 0.2 sec in process completion but NB takes 0 second.

Table 2. Evaluation on test split

Detailed	**J48**		**ID3**		**Naïve Bayes**	
Correctly classified Instances	61	100%	58	95.08%	61	100%
Incorrectly classified instances	0	0%	0	0%	0	0%
Kappa static	1		1		1	
Mean absolute error	0		0		0.00	
Root mean squared error	0		0		0.00	
Relative absolute error		0		0%		.27%
Root relative square error		0		0%		.84%
Time taken (second)		0.02		0.02		0
Total number of instances	61		61		61	

Table 3. Detailed Accuracy by Class

NAÏVE BAYES			**ID3**		**J48**	
DETAILED	BUG	NON BUG	BUG	NON BUG	BUG	NON BUG
TP RATE	1	1	1	1	1	1
FP RATE	0	0	0	0	0	0
PRECISION	1	1	1	1	1	1
RECALL	1	1	1	1	1	1
F-MEASURE	1	1	1	1	1	1
ROC	1	1	1	0.75	1	1

From table 3 it is clear that J48 give more correctly classified compare to ID3 and NB algorithms, with binary classification, precision as positive predictive value is the fraction of retrieved instances that are relevant while recall is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and measure of relevance but from the table 3 it is clear that recall-measure have equal value for corresponding three algorithms.

Table 4. Confusion Matrix

Confusion Matrix(J48)	**Confusion Matrix(ID3)**	**Confusion Matrix(Naïve Bayes)**
a b <-- classified as	a b <-- classified as	a b <-- classified as
55 0 | a = zero	55 0 | a = zero	55 0 | a = zero
0 6 | b = one	0 3| b = one	0 6 | b = one

From table 4 each column of matrix represents the instances in a predicted class, while each row represents in an actual class. Confusion matrix J48 shows 55 correctly classified and 0 none correctly classified another side 0 none correctly classified and 6 correctly classified arise. Confusion matrix in ID3 represents 55 correctly classified and 0 none correctly classified another side 0 none correctly classified and 3 correctly classified. In NB 55 correctly classified and 0 none correctly classified another side 0 none correctly classified and 6 correctly classified. It is clear from analysis correctly classified in J48 is total instances 61 is better value without error compare to other ID3 and NB algorithms.

4. CONCLUSION

In Weka all data is considered as instances attributes in the data for easier analysis and evaluation. Similar result is partitioned into several sub items. In the first part correctly classified instances will be partition in to numeric and percentage value, kappa statics, mean absolute error and root mean square error will be at numeric value only ID3 andJ48 time taken to build model: 0.2 seconds and test mode :10 fold cross validation. Here Weka compare all required parameters on given instances with the classifiers respective accuracy and prediction rate. Based on table 2 it can clearly see that highest accuracy of J48 is 100% without error also Naïve Bayes 100% correctly classified but with some error and ID3 95% correctly classified, so it is clear that J48 is the best in three respective algorithms so it is more accurate.

5. REFERENCES

[1] Hampherey Watts S., "A discipline for software Engineering reading", Ma,Addison Wesley,1995.

[2] Sunita Tiwari and Neha Chaudhary, "Data mining and Warehousing" Dhanpati Rai and Co.(P) Ltd. First Edition: 2010.

[3] J.R.Quinlan, "C4.5: programs for machine learning", Morgan Kaufmann,San Francisco,1993.

[4] 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.

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

[6] 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 Netherlands, 1995.

[7] Pal A. K., and Pal S., "Analysis and Mining of Educational Data for Predicting the Performance of Students", (IJECCE) International Journal of Electronics Communication and Computer Engineering, Vol. 4, Issue 5, pp. 1560-1565, ISSN: 2278-4209, 2013.

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

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

[10] 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.

[11] 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.

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

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

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

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

[16] Kotsiantis and Kanellopoulos, "Associationn rule mining: A recent overview", GESTS international transaction on computer science and Engineering, 2006.

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

[18] Fayyad, Piatesky Shapiro, Smuth and Uthurusamy, "Advances in knowledge discovery And data mining", AAAI Press,1996.

[19] Pal S., "Mining Educational Data to Reduce Dropout Rates of Engineering Students", I.J. Information Engineering and Electronic Business (IJIEEB), Vol. 4, No. 2, 2012, pp. 1-7.

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

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

[22] Vishal and Gurpreet, "A survey of text mining Techniques and applications", journal of engineering Technologies in web intelligence, 2009.

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

[24] Yadav S. K. and Pal S., "Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification", World of Computer Science and Information Technology (WCSIT), 2(2), 51-56, 2012.