Volume 1, Issue 3, July 2014 ISSN: 2348-8565 (Online)

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Discover the Rules by Different Machine Learning Algorithms in Software Bug Report during Project Development Process

Dhyan Chandra Yadav
Research scholar, Ph.D program in Computer
Shri Venkateshwara University
Gajraula, Amroha (U.P.), India
Email: dc9532105114@gmail.com

Rajeev Kumar
Assistant Professor,
Department of Computer Science.
Shri Venkateshwara University
Gajraula, Amroha (U.P.), India
E-mail: rajeev2009mca@gmail.com

ABSTRACT

Knowledge or technical skills play a major role in software engineering and repair and completion are important elements in the design of a software project because without these things engineers cannot make the required software. All software has different quality requirements for the software project but often some problems arise from the compatible software and its scripts. The above feature of the software is known as bug and is detected by the software tracking system. The bug fix problem is a noticeable problem and is advised in the repair problem report. We analyzed the issues and provided the relationship between the data object. In this paper we proposed and found interactions between the data object and found the percentage of confidence, support, data object with the help of apriori, predictive apriori and filtered algorithms.

Keywords:— *KDD, Association Rules, Apriori, Predictive Apriori and Filtered Associate, Weka.*

I. INTRODUCTION

Zach Barnett [1] introduced that a particular software engineer is malicious or has made certain software errors as a bug or bug fixer. Error is a major problem that often arises from the display of codes and the available information does not fully support the corresponding documents. There are many risk factors for the body of the software industry. The software industry is in dire need of quality, cost and time for customer satisfaction. Data mining provides a useful way and relationship between a data object to detect an error.

Data Mining is a process presented by Han, Kamber and Pei [2]. The whole point of algorithm and data mining is to extract useful data from a large amount of data. For example download all the information about a bug detection software project. Data Mining determines practical and audible structures, frameworks and movements by exploring data construction technologies such as machine learning and relevant algorithms and it is argued that data mining robots access data analysis and can detect significant data formation. Data mining has progressed to a more dynamic field of research for theoretical testing and application of problem-related applications. KDD's development incorporates all software project information and includes popular evidence and provides a link between the data object.

1.1 Association

Sunita Tiwari and Neha Chaudhary [3] introduced the law of organization. Integration provides relationships between data in any database. For example:

- A customer who purchases a compatible hard disk operating system.
- OS => hard disk [5% support, 70% confidence]
- Here is support and confidence the two-point meeting point below-
- Support 5% transaction under analysis so that the app and hard disk are purchased together.
- Confidence 70% - 70% customer purchasing operating system and purchasing a hard disk, here support (s) --Shows recent performance.
- Indicate whether the customer contains the application and contains the hard disk

1.2 Apriori Algorithm

Tiwari and Chaudhary have argued that Apriori is a Latin word and its meaning is "derived from what comes first" apriori using a lesser technique. The most popular and ancient algorithm for common pattern mining. This algorithm works with category attribute function.

Apriori is an organizational law algorithm proposed by Agrawal and Srikant in 1993 [4]. The law of the organization Apriori is used to dig up common patterns in the database. Support and self-confidence is a common method used to measure the quality of federal law. Example: The Apriori organization manages the common patterns mine in the bug and non-bug software database database and provides support and trust transaction support for a specific quality of organizational law.

1.3 Predictive Apriori Associative

Put [5] in the Apriori organization's prediction law algorithms, support and self-confidence have become one part called the authenticity of the analysis used to reproduce the apriori organization's law. It sets the stage for production, n the law of the supreme organization established on n appointed by the provider. Example: Guess (support, confidence) of software feature features and redesign of the organization's top production law.

1.4 Filtered Associator

Weka represents a filtered link for filtering training data. This algorithm is a random time sequence of data allowed to complete a random filter. The design of the filter is based entirely on the data of the preparation and the test conditions will be managed by the filter without changing its structure. Here in this algorithm we can look at the apriori, predictive apriori and filtered algorithm filtered to get the result.

II. Related Work

Shepperd, Schofield and Kitchenham [6] examined the need to assess the expenses of the executives and programming improvement associations and gave an idea of pattern and assessment techniques.

Alsmadi and Magel [7] examined how information digging gives space to another product venture with its quality, cost and multifaceted nature additionally making a channel between information mining and programming designing.

Boehm, Clark, Horowitz, Madachy, Shelby and Westland [8] talked about that some product organizations are encountering some precision issues relying upon his informational collection after the anticipating programming organization offered a novel plan to characterize a task cost plan and decide a staff time table.

K.Ribu [9] talked about the requirement for open source code extends that are broke down by anticipating and finding a product venture that centers around the article with case models.

Nagwani and Verma [10] examined that the mistake of the product (bug) and the normal bug a lot rating in all product synopses, with information mining additionally talked about the product bug.

Hassan [11] contended that a mind boggling information source (sound, video, text, and so forth) requires a great deal of cradle to measure and doesn't uphold the ordinary size and length of the cushion.

Li and Reformate [12] examined that .the product arrangement the executives framework incorporates reports, programming code, the norm estimation, mistake following plan model and incorporates update information.

Elcan [13] contended that the COCOMO model is subverted by precise quotes and there is a ton about cost assessment on the grounds that in venture development it includes a ton of adaptability and in this manner a COCOMO gauge as expected and measurements exertion.

Chang and Chu [14] talked about that to locate the primary information base example and its disparities and the connection between them in the information mining affiliation association.

Kotsiantis and Kanellopoulos [15] examined this significant defect in the improvement of a product venture and furthermore talked about the arrangement of assets for anticipating and administration to lessen the quantity of exchanges in the information base.

Pannurat, N.Kerdprasop and K.Kerdprasop [16] talked about that hierarchical law accommodates the connection between huge information bases, for example, programming venture time, cost record and aids venture improvement.

Fayyad, PiateskyShapiro, Smuth and Uthurusamy [17] examined that partition makes a relationship or guide between the information object and the classes portrayed before.

Harsh and Vassillios [18] contended that in the investigation of the mix of a similar article set in a similar gathering it likewise arranged the quality in the gathering so the contrasts between the gatherings were expanded comparable to the distinctions inside the gatherings.

Runeson and Nyholm [19] have contended that code replication is an autonomous language issue. Over and over another report of an issue in programming improvement and duplication emerges utilizing a neural language with an information mine.

Vishal and Gurpreet [20] contended that the information mine dissects information and investigates concealed information from the content in the improvement of a product venture.

Lovedeep and Arti information mining [21] offers a specific field of programming designing where numerous capacities are effortlessly worked with the best quality and lessen the expense and issues of prominent.

Nayak and Qiu [22] talked about that frequently as expected and cost, related issues emerge from programming venture improvement these issues distinguished in the difficult report, information mining gives assistance in lessening issues and disconnecting and decreasing other programming related bugs.

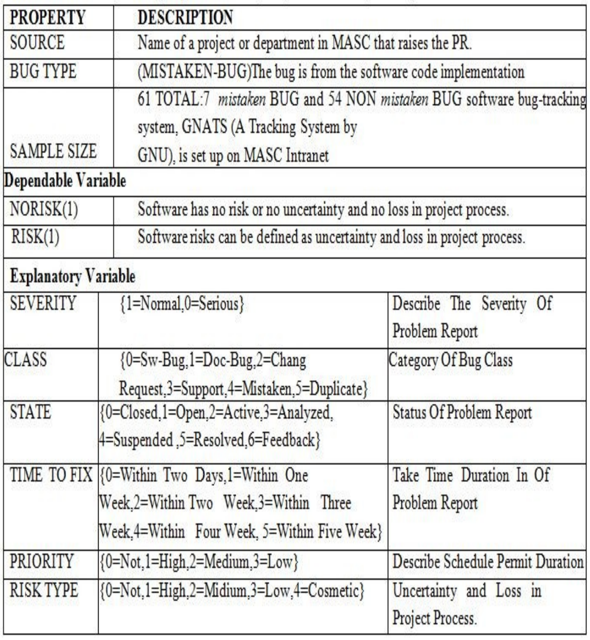
III. METHODOLOGY

In this paper we find out the best combination of different attributes in any data set .We also consider similarity and comparison the result of tree association algorithm: Apriori association rule, Predictive apriori association rule and Filtered associator rule.

3.1 Data Preparation

Error in problem report are often corrected. All problem reports are often divided into two categories: risk and risk. in danger a mistake may occur automatically within the software. Software tracking software, GANTS which may be a software bug tracking program. Error-error is a component of the category field in code usage. Now the error-correcting process is now performed using several standard processing operations, data processing, collection, classification, merging and operations got to be done. The database is made into the MS-Excel, MS word 2010 database and data management system to store data collection.

Table 1. Mistaken Bug Representation by Dependable & Explanatory Variables



PROPERTY	DESCRIPTION	
SOURCE	Name of a project or department in MASC that raises the PR.	
BUG TYPE	(MISTAKEN-BUG)The bug is from the software code implementation	
SAMPLE SIZE	61 TOTAL:7 *mistaken* BUG and 54 NON *mistaken* BUG software bug-tracking system, GNATS (A Tracking System by GNU), is set up on MASC Intranet	
Dependable Variable		
NORISK(1)	Software has no risk or no uncertainty and no loss in project process.	
RISK(1)	Software risks can be defined as uncertainty and loss in project process.	
Explanatory Variable		
SEVERITY	{1=Normal,0=Serious}	Describe The Severity Of Problem Report
CLASS	{0=Sw-Bug,1=Doc-Bug,2=Chang Request,3=Support,4=Mistaken,5=Duplicate}	Category Of Bug Class
STATE	{0=Closed,1=Open,2=Active,3=Analyzed, 4=Suspended ,5=Resolved,6=Feedback}	Status Of Problem Report
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 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}	Uncertainty and Loss in Project Process.

Data is generated consistent with the specified format and formats and therefore the data is converted to ARFF (affiliate file format) for processing to line. The ARFF file is an ASCII document that describes an inventory of situations that share a group of symbols.

3.2 Data Selection And Transform

In this step only those fields were selected which were required for data processing. A couple of derived variables were selected. Where a number of the knowledge for the variables was extracted from the database. All the predictor and response variables which were derived from database are given in table 1 for reference. The survey uses status of problem report analysis /non analysis and therefore 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 for the present investigation as follow:

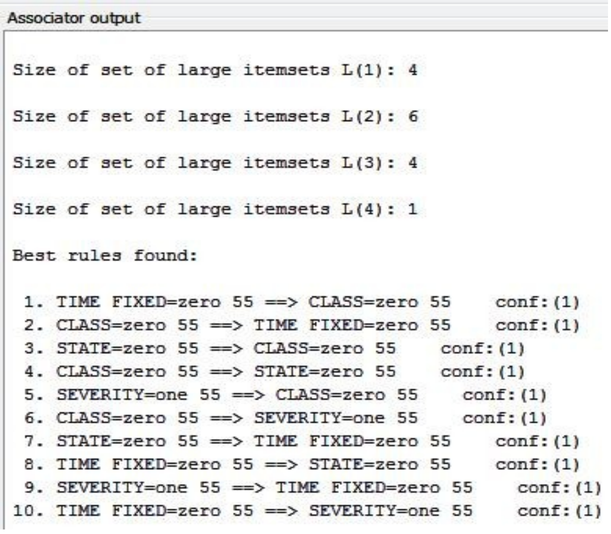
Working of Apriori algorithm using data mining tool software risk management contain large dataset, as the population increases day by day.

3.3 Data Mining Implementation

Here we look at the three algorithms of the merger law namely the apriori association affiliation law, the filter association, association law, the apriori association law. We compare the results of these integration programs with a data mining help tool. The

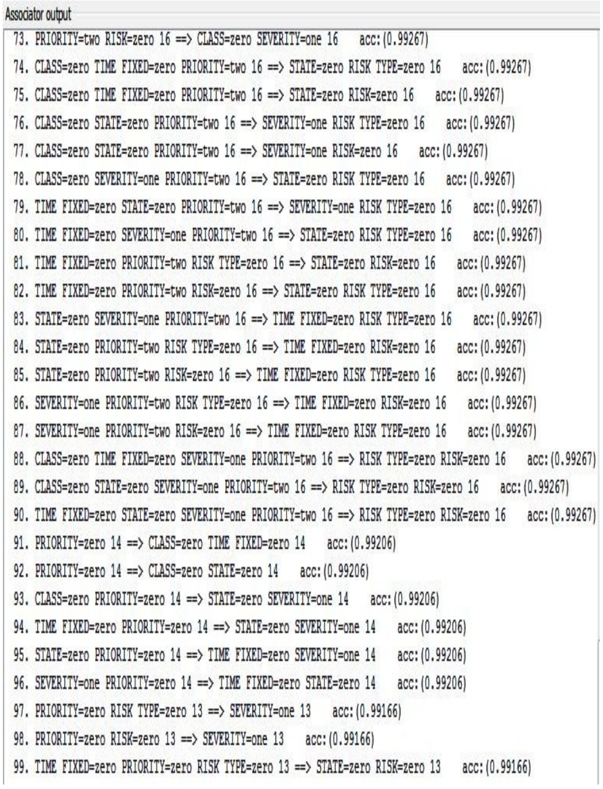
Apriori algorithm is described in various categories. Predictive organization control and confidence predictions are grouped into a single unit called predictive accuracy. In this precision prediction is used to produce the law of the apriori organization.

IV. Result And Discussion



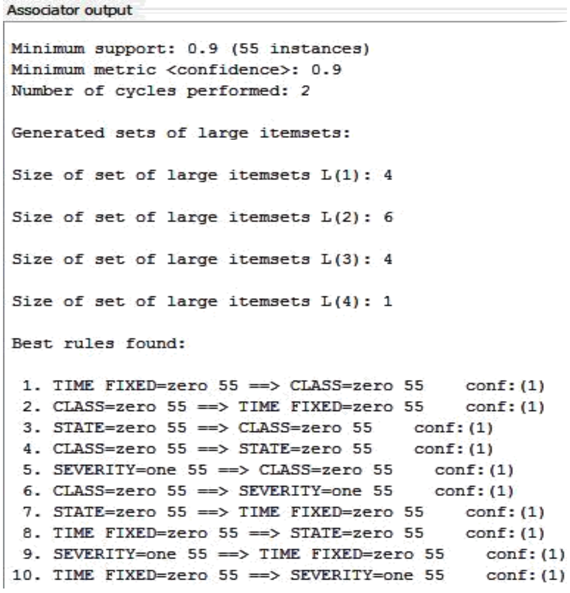
```
Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 6

Size of set of large itemsets L(3): 4

Size of set of large itemsets L(4): 1

Best rules found:

 1. TIME FIXED=zero 55 ==> CLASS=zero 55    conf:(1)
 2. CLASS=zero 55 ==> TIME FIXED=zero 55    conf:(1)
 3. STATE=zero 55 ==> CLASS=zero 55    conf:(1)
 4. CLASS=zero 55 ==> STATE=zero 55    conf:(1)
 5. SEVERITY=one 55 ==> CLASS=zero 55    conf:(1)
 6. CLASS=zero 55 ==> SEVERITY=one 55    conf:(1)
 7. STATE=zero 55 ==> TIME FIXED=zero 55    conf:(1)
 8. TIME FIXED=zero 55 ==> STATE=zero 55    conf:(1)
 9. SEVERITY=one 55 ==> TIME FIXED=zero 55    conf:(1)
10. TIME FIXED=zero 55 ==> SEVERITY=one 55    conf:(1)
```

Figure.1. Representation of Apriori In Weka



```
73. PRIORITY=two RISK=zero 16 ==> CLASS=zero SEVERITY=one 16    acc:(0.99267)
74. CLASS=zero TIME FIXED=zero PRIORITY=two 16 ==> STATE=zero RISK TYPE=zero 16    acc:(0.99267)
75. CLASS=zero TIME FIXED=zero PRIORITY=two 16 ==> STATE=zero RISK=zero 16    acc:(0.99267)
76. CLASS=zero STATE=zero PRIORITY=two 16 ==> SEVERITY=one RISK TYPE=zero 16    acc:(0.99267)
77. CLASS=zero STATE=zero PRIORITY=two 16 ==> SEVERITY=one RISK=zero 16    acc:(0.99267)
78. CLASS=zero SEVERITY=one PRIORITY=two 16 ==> STATE=zero RISK TYPE=zero 16    acc:(0.99267)
79. TIME FIXED=zero STATE=zero PRIORITY=two 16 ==> SEVERITY=one RISK TYPE=zero 16    acc:(0.99267)
80. TIME FIXED=zero SEVERITY=one PRIORITY=two 16 ==> STATE=zero RISK TYPE=zero 16    acc:(0.99267)
81. TIME FIXED=zero PRIORITY=two RISK TYPE=zero 16 ==> STATE=zero RISK=zero 16    acc:(0.99267)
82. TIME FIXED=zero PRIORITY=two RISK=zero 16 ==> STATE=zero RISK TYPE=zero 16    acc:(0.99267)
83. STATE=zero SEVERITY=one PRIORITY=two 16 ==> TIME FIXED=zero RISK TYPE=zero 16    acc:(0.99267)
84. STATE=zero PRIORITY=two RISK TYPE=zero 16 ==> TIME FIXED=zero RISK=zero 16    acc:(0.99267)
85. STATE=zero PRIORITY=two RISK=zero 16 ==> TIME FIXED=zero RISK TYPE=zero 16    acc:(0.99267)
86. SEVERITY=one PRIORITY=two RISK TYPE=zero 16 ==> TIME FIXED=zero RISK=zero 16    acc:(0.99267)
87. SEVERITY=one PRIORITY=two RISK=zero 16 ==> TIME FIXED=zero RISK TYPE=zero 16    acc:(0.99267)
88. CLASS=zero TIME FIXED=zero SEVERITY=one PRIORITY=two 16 ==> RISK TYPE=zero RISK=zero 16    acc:(0.99267)
89. CLASS=zero STATE=zero SEVERITY=one PRIORITY=two 16 ==> RISK TYPE=zero RISK=zero 16    acc:(0.99267)
90. TIME FIXED=zero STATE=zero SEVERITY=one PRIORITY=two 16 ==> RISK TYPE=zero RISK=zero 16    acc:(0.99267)
91. PRIORITY=zero 14 ==> CLASS=zero TIME FIXED=zero 14    acc:(0.99206)
92. PRIORITY=zero 14 ==> CLASS=zero STATE=zero 14    acc:(0.99206)
93. CLASS=zero PRIORITY=zero 14 ==> STATE=zero SEVERITY=one 14    acc:(0.99206)
94. TIME FIXED=zero PRIORITY=zero 14 ==> STATE=zero SEVERITY=one 14    acc:(0.99206)
95. STATE=zero PRIORITY=zero 14 ==> TIME FIXED=zero SEVERITY=one 14    acc:(0.99206)
96. SEVERITY=one PRIORITY=zero 14 ==> TIME FIXED=zero STATE=zero 14    acc:(0.99206)
97. PRIORITY=zero RISK TYPE=zero 13 ==> SEVERITY=one 13    acc:(0.99166)
98. PRIORITY=zero RISK=zero 13 ==> SEVERITY=one 13    acc:(0.99166)
99. TIME FIXED=zero PRIORITY=zero RISK TYPE=zero 13 ==> STATE=zero RISK=zero 13    acc:(0.99166)
```

Figure 2: Representation of Pridective Apriori



```
Minimum support: 0.9 (55 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 2

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 6

Size of set of large itemsets L(3): 4

Size of set of large itemsets L(4): 1

Best rules found:

 1. TIME FIXED=zero 55 ==> CLASS=zero 55    conf:(1)
 2. CLASS=zero 55 ==> TIME FIXED=zero 55    conf:(1)
 3. STATE=zero 55 ==> CLASS=zero 55    conf:(1)
 4. CLASS=zero 55 ==> STATE=zero 55    conf:(1)
 5. SEVERITY=one 55 ==> CLASS=zero 55    conf:(1)
 6. CLASS=zero 55 ==> SEVERITY=one 55    conf:(1)
 7. STATE=zero 55 ==> TIME FIXED=zero 55    conf:(1)
 8. TIME FIXED=zero 55 ==> STATE=zero 55    conf:(1)
 9. SEVERITY=one 55 ==> TIME FIXED=zero 55    conf:(1)
10. TIME FIXED=zero 55 ==> SEVERITY=one 55    conf:(1)
```

Figure 3: Representation of Filtered Associator

The apriori algorithm functionality uses a data mining tool where Figure 1 Demonstrates how to import a database into this tool., Figure 2 and shows the previous creation of a data domain in this Figure 3. To the left a list of database attributes is provided. We have created an arff file containing numerical attributes and a numerical filter and a title from the weed used to convert numeric attributes into a name. The database is filtered and subjected to algorithms for various organizational rules.

We find the results from figure 1, figure 2 and figure 3 using these three association rule algorithms. In this case we use a bug data collection to manage software risk and compare 7 tokens and 61 case records. Find organization rules using variables in computer process by weka and represent the result using apriori, predictive apriori and filtered algorithm.

V. Conclusion

Association rules algorithms are used to produce the same rules for bug-bug risk detection and perform software risk management. Test results show that algorithms namely apriori, predictive apriori and filtered composite work well. The TOP TEN rules made by algorithms are the same and important for these rules the investigator takes help in distinguishing the software bug from software risk management.

So these algorithms can be used in other domains to bring attraction between existing data to the root. The organization rules produced by these three algorithms can be combined to get the best results for any real-life application. Algorithms can also be integrated into an effective algorithm.

REFERENCES:

[1] Zach Barnett "A senseless conversation "May 2012.

[2] [2].Jiawei Han, Micheline Kamber and Jian Pei "Data Mining Concepts and Techniques" Morgan Kaufmann Publishers Third Edition. 2012.

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

[4] [4].Agrawal, R., Imielinski, T., and Swami, A. N., "Mining association rules between sets of items in large databases", In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216, 1993.

[5] http://www.cs.waikato.ac.nz/ml/weka/.

[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 Westl and, "Cost models for future software life cycle Process: COCOMO2.0." in Annals of software Engineering special volume on software process and product measurement, J.D. Arther and S.M. Henry, Eds, vol.1, pp.45-60,j.c. Baltzer AG, science publishers, Amsterdam, The Netherlands, 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: Arecentover view", GESTS international transaction a 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 computer 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", in proceeding 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 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.

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

* * * * *