RESEARCH ARTICLE OPEN ACCESS

A Review on Software Mining: Current Trends and Methodologies

Gurtej Singh Ubhi
M.Tech Scholar
Lovely Professional University

Jaspreet Kaur Sahiwal
Assistant Professor
Lovely Professional University

ABSTRACT
With the evolution of Software Mining, it has enabled to play a crucial role in the day to day activities .By empowering the software with the data mining methods it aids all the software developers and the managers at a the managerial levels to use it as a tool for knowledge discovery from the relative software engineering data (in the form of code, design documents, bug reports) that would help in visualizing the project's status of evolution and progress. To add on, further all the mining methods and algorithms help to device the models that would result in the development of any fault prone real time system prior to the testing phase or the evolution phase. This review paper will highlight the different methodologies for software mining along with its extension as a tool for fault tolerance and bibliographic reference with the special eminence on mining the software engineering information.
Index Terms: detection strategies, prediction strategy, fault, metrics thresholds, software metrics, software fault prediction, software quality, MUD(mining Unstructured Data)

I. OVERVIEW

It is found that a lot of software institutions have gathered huge volumes of data with the hope of the better understanding and evaluation of their processes as well as their products. Although useful in extraction of larger set of data but at the same time plethora of valid and useful data remains hidden in the software engineering data warehouse and data engines along with the software repositories. To answer all those questions, the software mining has emerged as a tool for the better exploration of all such software engineering data.

II. INTRODUCTION

By definition Software mining means the process of the extraction of novice, minute data with all the best possible information from the software engineering repositories. At the same time this marks a shift in the approach from the verification driven methodology to discovery data driven approach. It should be noted that the discovery driven approach is not a new methodology but its popularity is dedicated to the following characteristics.

1. Mature enough to handle intensive computations.
2. Cheap data storage methods and its ability to integrate with the large amount of readily available data.
3. Advancements in the techniques like blueprint recognition, neural networks and decision trees.

Many studies have shown the usage of this kind of data i.e. the software engineering data supports the diverse aspect of the software maturity within the built-up and other sources of utilities. Given that the idea of application of basic data mining modus operandi to the software engineering records has exist since middle of 1990s [1], the initiative has created the revolution.

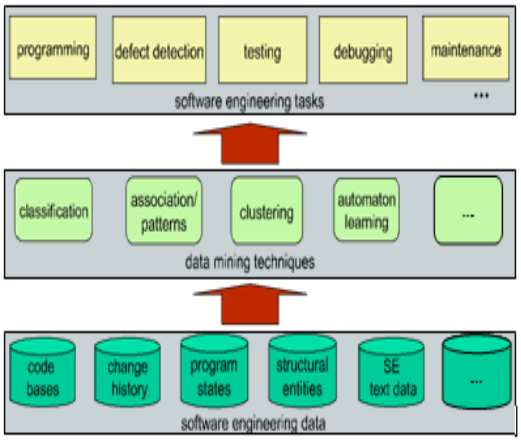

Figure 2.1. General idea of mining SE data

III. SOME SPECIAL ISSUES

As per the researchers, they have explored a variety of software engineering based questions using the software repository data by using it as primary tool for the source of information. The major commonly explored areas include the process of the software evolution , modeling of the software development process along with the usage

of the machine learning , prediction of the future based on the software quality , software bug analysis , software pattern change and evaluation of the software clones.

IV. HUMAN ASPECT OF SOFTWARE ENGINEERING

The central question in how to improvise the software art centers, as it always has, on people [2]. Authors in [3] have shown and developed the potential ripeness sculpt that states the issues akin to job recital, lineup solidity in addition to listed all the reasons that are the key points for the software success. It is therefore, utmost to ponder more on the human based facet of software engineering for the better software triumph.

V. REVIEW OF LITERATURE

5.1 A New Adaptive Architecture [6]

It will be a better idea to use both of the algorithms collectively. Apriori algorithm is good for small datasets where the minimum number of candidates have been generated. And FP-Tree algorithm is better for large databases where data are huge in numbers. This new approach will help data mining tasks in software engineering much easier. A threshold or limit can be set for transactions. If transactions exceed the limit then Fp-Tree should be used and anything below that limit will ultimately lead towards the Apriori. The proposed architecture in Fig.5.1 can be easily embedded in the previous work by[4] [5].

This includes two steps.

1. All the set of laws in the way of the jeopardy factor and their mitigations have been recorded in the familiarity pedestal.
2. Well-built relations between the risk factors would be generated using both algorithms.

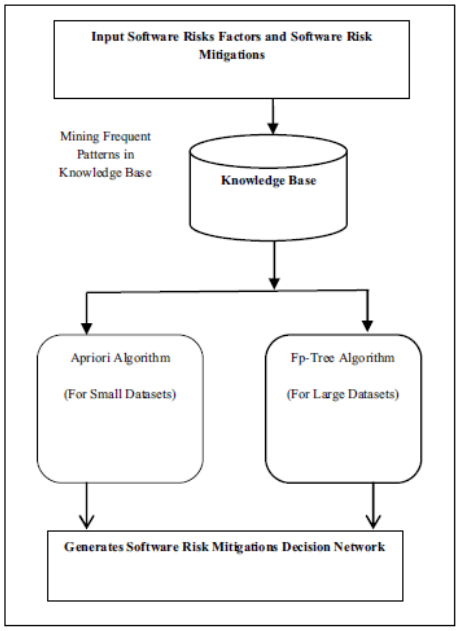

Figure 5.1. A New Adaptive Architecture.

5.2 Mining Unstructured Data In Software Repositories: Current And Future Trends [7]

The Unstructured data is defined as the data that is not having the principles of organization, schema or the structure. Such kind of the data may include (email messages, all kinds of software documentations etc.). It is found that the accessibility of unstructured data for the entire software engineer has seen a sharp growth. Following methods may be employed to undergo mining of such data:

1. Pattern Matching.

This can be done by following steps.

Step 1. Identify set of the documents likely related to the specific document.

Step 2. Trace the related artifacts that contain the external links as prepared by bacchelli al. [8] to identify all the mail messages defined.

2. Natural Language Processing

This allows extracting the piece of meaningful information from all the text written in the natural language. This includes

- Text Summarization.
- Auto completion of text.
- Sentiment Analysis of Polarity.
- Topic Segmentation of cohesive segments.

5.2.1 Future of MUD

The future of mining unstructured Data (software engineering data) is helpful in following ways.

- As a tool to enhance Qualitative Analysis.
- Crosscutting Analysis.
- Inter-Domain Recommendation.
- Mining Multimedia Contents.

5.3 Text Mining And Software Engineering : An Integrated Source Code And Document Analysis Approach [9]

This approach is beneficial as with eternally escalating number of the computers and their computations in on increase at a alarming rate , it is found that an estimate of 250 billion lines of code were being uphold in 2000 and it is still increasing [10]. This makes the use of the Ontology based program comprehension way. As with the software age, the crucial task of maintenance of software is becoming more complex as well as crucial. Software evolution often regarded as software maintenance has the preponderance of the entirety price tag that occurs through the life extent of the software organization [10, 11]. The safeguarding challenges is a result of the different inter-relationships and the representations the exist among the all software knowledge resources [12, 13]. An vital ingredient of our construction is a software ontology that capture major concept and dealings in the software safeguarding sphere of influence. Following picture shows the integrated approach.

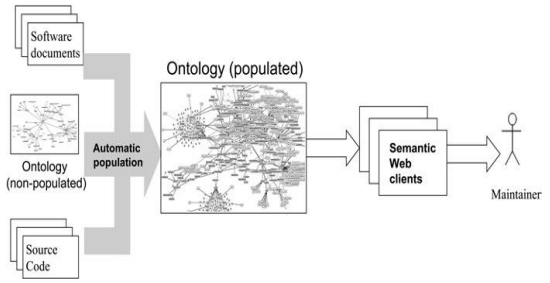

Figure 5.2. Ontological text mining of software documents for software engineering.

5.4 The Web Software Mining Based On Vector Space Model [14]

This uses the methodology similar to a web based crawler that is designed and implemented according to the features of the World Wide Web. The procedure is as follows.

Step 1. URL collector collects the hyper texts and then the software downloads all the addressed via the http protocols dynamically.

Step 2. The collected pages are then cleaned and filtrated.

Step 3. The validity of the software addresses is then checked and they are saved into the information database.

Step 4. Text Classification library is then established.

Step 5.From the target sample the feature information is then extracted.

Step 6.Returning the information that matches the threshold conditions.

Step 7. Then finally, the software description text is classified which is supposed to be stored in the information database.

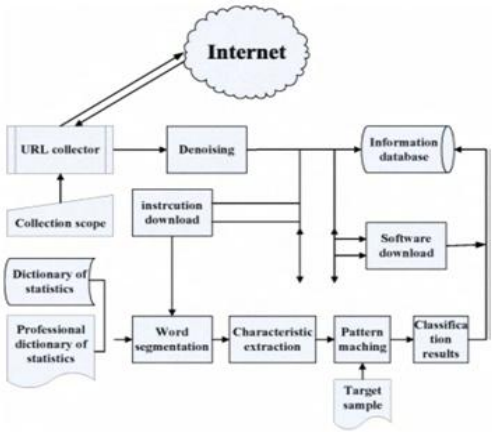

Figure 5.3. Structure of software mining system.

5.5 Software Development Process Mining[15]

Due to dependence of the software in our on a daily life, the development of the same has been a crucial part in our lives.But at the same time the process is not formalized which means the model to it is not available. This uses the following set of methodologies.

- Process delivery in the Software development.
- Conformance Checking
- Process Enhancement

Hypothesis taken into picture while the process delivery is done which are as:

- [H01a] Software expansion process exposed from dealings testimony from enlargement tackle used are inadequately comprehensive and perfect to be useful for software trade purpose.
- [H01b] It is not feasible to endow with criticism to agile developers in real instance

vis-à-vis the course being executed, explicitly by being able to distinguish the part of each squad affiliate.

Hypothesis taken into picture while the conformance checking is done which are as:

- [H02a] It is not doable to recognize divergence Between what agile developers do in put into observe and what they were hypothetical to do.

- [H02b] There is no noteworthy unpredictability in the roles perform surrounded by an improvement team in nimble approach.

Hypothesis taken into picture while the process enhancement is done which are as:

- [H03a] It is not potential to endow with criticism to the agile developer, concerning the superiority of the process he/she is executing.
- [H03b] It is not probable to mechanically adapt the IDE to get better the developer's recital, based on the scrutiny of the IDE event firewood.

5.6 MDA Based Approach for Software Mining System. [16]

This technique uses the concept of the constraints and various set of architectures .Further it is based on the framework of the UML that extends the user-interface extension views. This is further divided into following set of methodologies:

- System Architecture Modelling
- Component Modelling

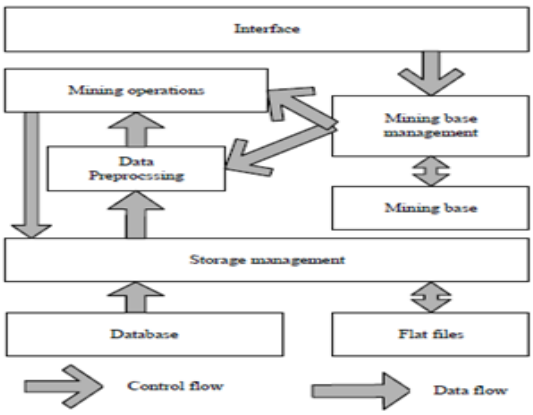

Figure 5.4. A outline of objective platform model for software mining system.

VI. A COMPARATIVE STUDY OF VARIOUS TOOLS AIDED FOR SOFTWARE ENGINEERING.

6.1 *CGMINER*

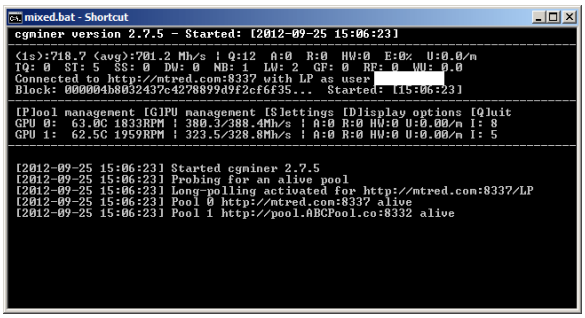

Figure 6.1. A CGMiner tool

The best part of CGminer is that is written in C language that makes it usable on wide variety of platforms like Windows, Linux etc. The other features includes over-clocking and monitoring interface capabilities.

Application Area: Self detection of mini databases along with binary block of the kernels makes it much usable for vector based software mining activities.

6.2 BFGMiner

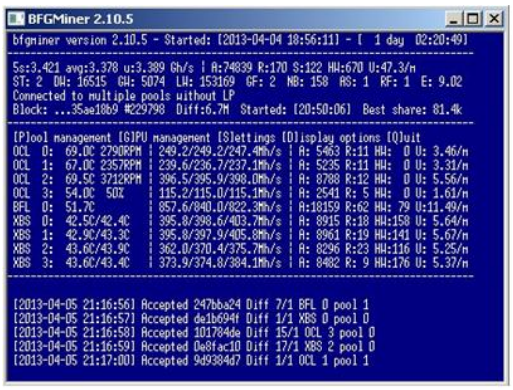

Figure6.2. A BFGMiner Tool

This miner tool is the further enhancement of the CGMiner tool that encapsulates the features like dynamic monitoring and interface capabilities.

Application Area: Although, it is a versatile program it has inbuilt vector support and fan control that makes it best suited for not only vector based mining scenarios but also for multidimensional unstructured software engineering data.

6.3 BITMINER

Figure 6.2. A BITMINER Tool

This tool is totally an advanced version of the new available tools because of the following facts.

1. Reduce stale work.
2. Assures good mining speed.
3. Having OpenCL-Compatible GPU's.
4. Long Pooling Methods.

Application Area: Useful for state full software engineering data where the degree of interactions is the most important element.

6.4 BTCMiner

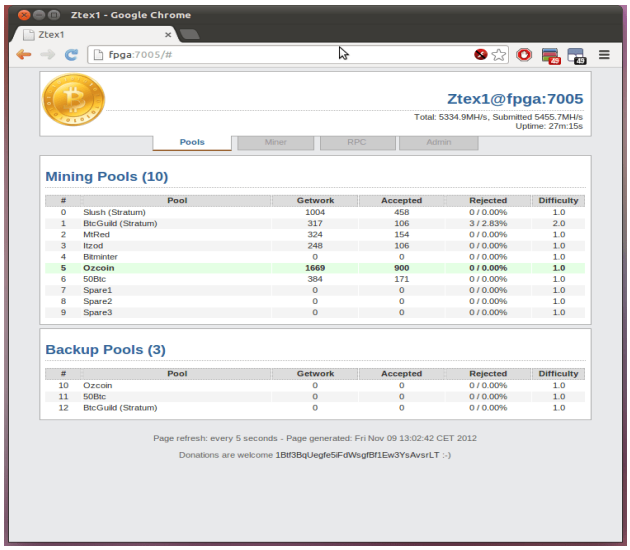

Figure 6.4 A BTCMiner tool

The following add on features makes it more useful.

- Dynamic frequency scaling.
- Better error measurement.
- Hash based mining methodology.
- Allows to run all kinds of mining threads from the single instances.

Application Area: This is quite helpful in areas where the efficiency factor is the most contributing for the calculation of mining strategy in scenario based usages.

6.5 POCLBM

Figure 6.5. A POCLBM tool

This is python based mining software that comes with the list of following features.

1. Enable to write codes that will work across a variety of programs.
2. Greater for experimentation.
3. Producing hash rates of higher magnitude.
4. Works perfectly with 4MD.

Application Area: As the name suggests it is quite helpful in the Ares with mining requires the mapping issues along with the heuristic machine learning methods.

6.6 DIABLO MINER

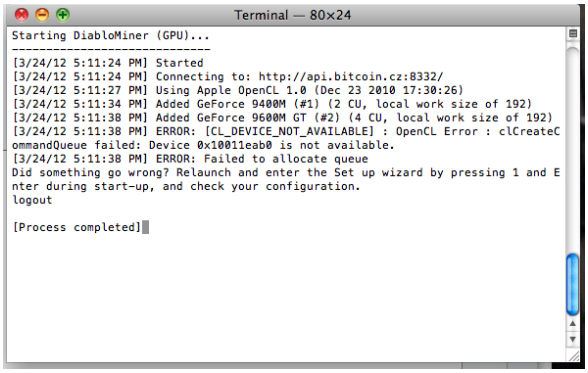

Figure 6.6. A DIABLO tool

Since it is java based tool, has the following set of features:

1. Better performance to hashing applications.
2. Supports unlimited pools.
3. Able to switch next pools if the connection failure occurs.

Application Area: This is quite useful in distributed software based mining based areas where the abstractions are of high usage like bitmaps, data localized in githubs.

VII. CONCLUSION

In this paper it has been presented that how the software mining methods has been evolved and making it as a tool for the purpose of the fault tolerance, better evaluation methods to rationalize the decisions for efficient quality based attributes and enabling to take better inferences from the mining methods evolved.

REFERENCES

[1] M. Mendonca and N. L. Sunderhaft. Mining software engineering data: A survey. A DACS state-of-the-art report, Data & Analysis Center for Software, Rome, NY, 1999.

[2] Brooks, F.P. Jr,1987, "No Silver Bullet: Essence and Accidents of Software Engineering." IEEE Computer 20(4) ,pp 10-19.

[3] Bate R.," A Systems Engineering Capability Maturity Model,Version 1.0", (CMU/SEI-94-HB-4, ADA293345).Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1994.

[4] Asif, M., Ahmed, J. and Hannan, A. "Software Risk Factors: A Survey and Software Risk Mitigation Intelligent Decision Network Using Rule Based Technique", An International Conference on Artificial Intelligence and Applications (ICAIA), The International MultiConference of Engineers and Computer Scientists (IMECS), Hong Kong, March 12-14, Vol. I, pp 25-39, 2014.

[5] Asif, M. and Ahmed, J. "Mining Frequent Patterns in Software Risk Mitigation Factors: Frequent Pattern-Tree Algorithm Tracing", International Conference on Machine Learning and Data Analysis (ICMLDA), World Congress on Engineering and Computer Science (WCECS), San Francisco, USA, October 22-24, 2014.

[6] Muhammad Asif and Jamil Ahmed :"Analysis of Effectiveness of Apriori and Frequent Pattern Tree Algorithm in Software Engineering Data Mining" ,IEEE, 2015 6th International Conference on Intelligent Systems, Modelling and Simulation.

[7] Gabriele Bavota .: "Mining Unstructured Data in Software Repositories:Current and Future Trends" in 23rd International Conference on Software Analysis, Evolution, and Reengineering , IEEE-2016.

[8] A. Bacchelli, M. Lanza, and R. Robbes. : "Linking e-mails and source code artifacts," in Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1, ICSE 2010, Cape Town, South Africa, 1-8 May 2010. ACM, 2010, pp. 375–384.

[9] R. Witte , Q. Li , Y. Zhang :" Text mining and software engineering: an integrated source code and document analysis approach", IEEE, 2008,p.3-16

[10] Sommerville, I.: 'Software engineering' (Addison-Wesley, 2000, 6th edn.)

[11] Seacord, R., Plakosh, D., and Lewis, G.: 'Modernizing legacy systems: software technologies, engineering processes, and business practices' 'SEI series in SE' (Addison-Wesley, 2003)

[12] Storey, M.A., Sim, S.E., and Wong, K.: 'A collaborative demonstration of reverse engineering tools', ACM SIGAPP Appl. Comput. Rev., 2002, 10, (1), p. 18–25

[13] Welty, C.: 'Augmenting abstract syntax trees for program understanding'. Proc. Int. Conf. Automated Software Engineering, 1997, (IEEE Comp. Soc. Press), pp. 126–133

[14] Feijie Wang , Zhongying Bai : "The Web Software Mining Based on Vector Space Model" ,(2009 First International Conference on Future Information Networks)

[15] João Caldeira,:" Software Development Process Mining: Discovery, Conformance Checking and Enhancement", (2016 10th International Conference on the Quality of Information and Communications Technology)

[16] Weichang Feng. :" MDA Based Development Approach for Software Mining System", (2010 Second International Workshop on Education Technology and Computer Science)