

# Implementing Of Data Mining Process by Means Software Engineering Approach

D.kiran kumar

*Research Scholar,  
SunRise University,  
IET Group, Alwar,*

Dr. Sudhir Dawra

*Professor in CSE  
SunRise University,  
IET Group, Alwar,*

Dr.K.Bhargavi

*Professor in CSE  
Palamur University,  
Andhra Pradesh.*

## ABSTRACT

Software organizations produce data in huge amount while creating and developing software. Each stage of software development witnesses a different set of data that gets produced right from the requirements phase till maintenance. Software organizations, in their continuous quest of improving the software quality put in considerable efforts to collect and maintain the data generated in software repositories. This huge data maintained in software repositories is analyzed using various Data Mining techniques which are used in extracting novel patterns and highlights from data. This area has attracted the attention of researchers in recent times and has become a preferred interdisciplinary research area of Software Engineering and Data Mining by researchers. In this review paper, the researcher tries to explore the various application areas of data mining in software engineering, different types of software engineering data where data mining is being and can be applied and various tools of data mining that are available and have been used by researchers to solve their related problems. Finally, based on this classification, the researcher tries to zero out the most important software engineering area that is garnering the researchers' attention.

**Keywords:** *Data mining, software engineering, hidden patterns, software engineering data, Software tools, software techniques.*

## 1. INTRODUCTION

The field of software engineering continues to remain the heartbeat for academic researchers from past several decades. The popularity of software engineering areas and their relevance has not reduced a bit, in fact, the onus on software developers to deliver a high quality software has increased with time in today's competitive world where software has become inseparable part of our lives in some form or the other.

Over the past few years, the focus of researchers & practitioners has shifted to interdisciplinary research areas and the most popular area in this context is the mix of Data mining and software engineering as Data mining is one such field which provides a lot of scope for research as the several techniques of It can be very useful in extracting novel & valuable information from the gathered data. It is the process of finding hidden patterns from the data stored in several databases which can be from any of the databases-operational, distributed, external, hypermedia databases. Several techniques available in Data Mining allow researchers to fetch novel patterns from data stored in these databases. This data extraction helps organizations to make business decisions.

Apart from these regular databases, a lot of data is produced during software development that can be used to extract valuable information using data mining techniques which will further improve the quality of software under development. This Software Engineering Data (SE Data) generated from various stages of software development is very useful and it is stored in what is known as Software Repositories.

The data in all these repositories becomes of a great value to software organizations because they realize the fact that by applying several data mining techniques, they can get great insights into the software under development. Popular Data Mining Techniques like clustering, which is an unsupervised learning technique and used for discovering groups & identifying interests & distribution, classification which is a supervised learning technique is used for mapping a data item into several predefined classes. Then there are techniques like Association rules, text mining, data characterization and change & deviation detection that can be used to mine different types of data available in several software repositories.

A lot of researchers have put considerable amount of efforts to categorize various Data Mining techniques that can be used in software engineering areas, some have tried to generalize various SE data on which such techniques may be applied to fetch valuable information, some have focused on various Data Mining tools that can be used to apply mining algorithms and techniques on SE data and so on. In other words, a lot of quality work has happened in the past in this very interesting interdisciplinary field.

However, the authors here feels that there is an immediate need to put all the work done by earlier researchers into a consistent format that covers all aspects ranging from various SE repositories, data that is available from SE repositories, data mining techniques available, various tools that are available and tools which are best to use on different kinds of mining techniques and on what kind of SE data such tools can be applied. In this review paper, the author presents an overview of the connected research areas of software engineering and data mining. The author tries to explore the various software engineering areas where data mining can be applied, and what tools should be applied on what kind of SE data and what tools are best for using a particular data mining technique. Finally, based on this classification, the author tries to identify the most important SE area which is garnering the most attention.

## II. SOFTWARE ENGINEERING DATA THAT CAN BE MINED

A software project that goes under development passes through different stages in its life cycle and generates a lot of data at every stage which can be structured or unstructured. All these different types of data can be mined using various mining techniques available.

## III. TOOLS USED IN DATA MINING

In this last section, we will present a summarized analysis of the tools that have been mostly used by the researchers according to the various SE areas discussed previously.

### Tools Summarization

Tool	Purpose	Where
Clone tracker	For mining code clones	Source code
Sqminer	to mine the sequences of changed-file and to process Subversion log entries.	Source code

CallExtractor	to analyze go programs. It extracts calls made to imported packages	Source code
J-REX	an optimized evolutionary code extractor,	Source code
FPClassifier	For training and classifying software modules	Source code
EvoONT	a software repository data exchange format based on the Web Ontology Language (OWL). EvoOnt includes software, release, and bug-related information.	Software repository mining projects
Isparql	Semantic Web query engine containing similarity joins for analyzing the extent of change between versions or to detect bad code smells.	Software repository mining projects
Deep	a tool for rehydrating evaporated information. It works as a Visual Studio plugin which	Source code,

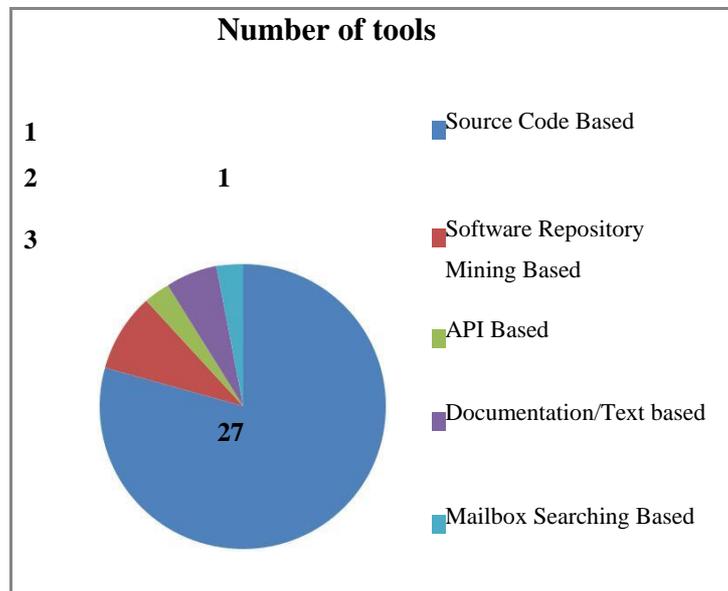
Intellisense	consolidates and presents historical information about code.	Documentation
SpotWeb	a web-based version of Spotnet and requires an existing HTTP server. Works on PHP5 (preferred is PHP 5.3 and up) to implement its functions, Compatible on Linux and Windows.	API
PopCon	POPcon™ Exchange POP3 Connector connects your Exchange server to your internet mailboxes. Over the time, it downloads emails from POP3 and IMAP mailboxes and distributes them to exchange mailboxes.	Emails, Mailbox searches
CP-Miner	A Tool for Finding Copy-paste and Related Bugs in Operating System Code which uses mining techniques to efficiently identify copy-pasted code in large software including operating systems, and detect copy-paste related bugs.	Source Code
PatchMiner	For frequent pattern mining, itemset mining	
Binary Analysis Tool	For analysis of binary files in source code	Source code
Checkstyle	Static code analysis tool to check java code for	Source code

	coding rules	
Apache Pig	Tool that work on Big data, with a provision of multiple query approach, that results in less number of data scan iterations.	Big Data
Mamoset	Used on software repository mining, source code to find new bug patterns.	Source code
Evolizer	a platform for mining software archives	Software repositories
ChangeDistiller	a tool for change extraction and analysis	Source code
ReSpam	Sequential mining on strings, pre-processing tool to convert texts to sequences of Integers	Text documents
SPIRIT	Sequential Pattern Mining with regular Expression constraint	
Exceptiontracer	an Eclipse plugin to resolve exceptions related to the stack trace in Java programs	
PR Miner	a tool that mines programming rules from source	Source code

	Code	
Perracotta	a dynamic analysis tool for Mining Temporal API Rules from Imperfect Traces	API
MAPO	Mining API Usages from Open Source Repositories	API, software Repositories
XSnippet	a tool for querying a sample repository for relevant code snippets for programming.	Source code
Strathcona	a system using multiple search to query for source code examples	Source code
BugLocator	an information retrieval based method for locating the relevant source code files for bug fixing based on an initial bug report.	Source code

BLIA	Bug Localization with Integrated Analysis. A Tool using stack traces in bug reports, information from source files, code change histories.	Source code
HotoshIO	attempts to overcome the problem of different data structures through a fuzzy equivalence matching. HitoshiIO compares the input/output of functions while observing their executions in-vivo.	
EQMINER	A C language tool that takes random input and dynamically executes relevant functions to detects clusters of similar code.	Source code
PR-MINER	Uses frequent itemset mining to extract implicit programming rules from large C code	Source code
CHRONICLE R	a tool supporting remote debugging. by capturing program execution sounds. It guarantees accurate replay in the lab, with very low overhead.	Source code
PROSPECTO R	Analyzes Python code and offers information about errors, violations and complexity.	Source code
DECKARD	a tree-based, scalable, and An accurate code clone detection tool which is scalable and tree based and has provisions for uncovering clone related bugs.	Source code
NICAD	Automated Detection of Near-Miss Intentional Clones , code clone detection using sequential pattern mining	Source code

Boa	A Large Scale Software Repository.	Source code
KEEL	Knowledge Extraction based on Evolutionary Learning)It is an open source Java tool that can be used for different knowledge data discovery tasks	Software Repositories
RapidMiner	an IDE for machine learning, data mining, text mining, predictive analytics and business analytics.	Text documents
SAS Text Miner	For text mining and analysis	Text documents
CCFinder	a C++ based token-based code clone analysis tool	Source code
SourceMonitor	A C++ tool to find out code quantity and modules complexity.	Source code
MALLET	A Java-based package for statistical natural language processing, document classification, clustering	Source code, text documents
CHANGEDIS TILLER	enables fine-grained change type extraction and analysis to reason about coding conventions, control or exception flow, and even code and comment coevolution.	Source code



**Fig 1: Pie Chart of Various Tools Categories**

It can be deduced from the above table that more than 75% of the tools and techniques developed and proposed by researchers focus on the source code while remaining other tools are based on other SE data like text documents, mailboxes, API's, other software repositories and projects.

## VI. CONCLUSION

In this paper, the researchers have tried to provide a three way analysis of various Software Engineering Data that are in focus, categorized the various types of tools and a summarized analysis of various Data Mining Tools and their focus area in Software Engineering data sources. The extensive literature survey and the tools analysis depicts that most of the work till today and most of the tools proposed by researchers till today has been on software code, thus there is a great scope of work in other areas of software engineering where data mining and its techniques can prove to be quite helpful in uncovering important patterns that may in turn assist developers, testers, maintenance teams and other people related to Software Engineering in some way or the other.

## REFERENCES

- [1]. M. Halkidi, D. Spinellis, G. Tsatsaronis *et al.*, "Data mining in software engineering," *Intelligent Data Analysis*, vol. 15, no. 3, pp. 413-441, 2011.
- [2]. A. E. Hassan, and R. C. Holt, "Predicting change propagation in software systems," in *Proceedings of the 20th IEEE International Conference on Software Maintenance*, 2004, pp. 284-293.
- [3]. Chaturvedi K.K, Singh V.B, Singh P, "Tools in Mining Software Repositories", 13th International Conference on Computational Science and Its Applications, pp. 89-98, 2013

- [4]. J. Huffman Hayes, A. Dekhtyar and J. Osborne, Improving requirements tracing via information retrieval. In *Proceedings of the International Conference on Requirements Engineering*, 2003.
- [5]. J. Huffman Hayes, A. Dekhtyar and S. Sundaram, Text mining for software engineering: How analyst feedback impacts final results. In *Proceedings of International Workshop on Mining Software Repositories (MSR)*, 2005.
- [6]. D. German and A. Mockus, Automating the measurement of open source projects. In *Proceedings of the 3rd Workshop on Open Source Software Engineering, 25th International Conference on Software Engineering (ICSE03)*, 2003.
- [7]. C. Jensen and W. Scacchi, Datamining for software process discovery in open source software development communities. In *Proceedings of International Workshop on Mining Software Repositories (MSR)*, 2004.
- [8]. C.C. Williams and J.K. Hollingsworth, Automating mining of source code repositories to improve bug finding techniques, *IEEE Transactions on Software Engineering* **31**(6) (2005), 466–480.
- [9]. S. Morisaki, A. Monden and T. Matsumura, Defect data analysis based on extended association rule mining. In *Proceedings of International Workshop on Mining Software Repositories (MSR)*, 2007.
- [10]. R. Chang, A. Podgurski and J. Yang, Discovering neglected conditions in software by mining dependence graphs, *IEEE Transactions on Software Engineering*, 2008.
- [11]. W. Dickinson, D. Leon and A. Podgurski, Finding failures by cluster analysis of execution profiles, *International Conference on Software Engineering (ICSE)*, 2001.
- [12]. M. Last, M. Friedman and A. Kandel, *The Data Mining Approach to Automated Software Testing*, In *Proceeding of the SIGKDD Conference*, 2005.
- [13]. J. Bowring, J. Rehg and M.J. Harrold, Active learning for automatic classification of software behavior, *International Symposium on Software Testing and Analysis (ISSTA)*, 2004.
- [14]. C. Liu, X. Yan, and J. Han. Mining control flow abnormality for logical errors. In *Proceedings of SIAM Data Mining Conference (SDM)*, 2006.
- [15]. C. Liu, X. Yan, H. Yu, J. Han and P. Yu, Mining behavior graphs for ‘backtrace’ of noncrash bugs. In *SIAM Data Mining Conference (SDM)*, 2005.
- [16]. Y. Kannelopoulos, Y. Dimopoulos, C. Tjortjis and C. Makris, Mining source code elements for comprehending object oriented systems and evaluating their maintainability, *SIGKDD Explorations* **8**(1), 2006.
- [17]. D. Engler, D. Chen, S. Hallem *et al.*, “Bugs as deviant behavior: A general approach to inferring errors in systems code,” *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 57-72, 2001.
- [18]. Z. Li, and Y. Zhou, “PR-Miner: Automatically extracting implicit programming rules and detecting violations in large software code,” in *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering*, 2005, pp. 306-315.
- [19]. S. Lu, S. Park, C. Hu *et al.*, “MUVI: automatically inferring multi-variable access correlations and detecting related semantic and concurrency bugs,” *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 103-116, 2007.
- [20]. B. Baker, “On finding duplication and near-duplication in large software systems,” in *Second IEEE Working Conf on Reverse Eng. (wcre)*, 1995, pp. 86-95.
- [21]. T. Kamiya, S. Kusumoto, and K. Inoue, “CCFinder: a multilinguistic token-based code clone detection system for large scale source code,” *IEEE Transactions on Software Engineering*, pp. 654-670, 2002.
- [22]. V. Wahler, D. Seipel, J. Wolff *et al.*, “Clone detection in source code by frequent itemset techniques,” in *Fourth IEEE International Workshop on Source Code Analysis and Manipulation*, 2004, pp. 128-135.



- [23]. W. Qu, Y. Jia, and M. Jiang, "Pattern mining of cloned codes in software systems," *Information Sciences*, 2010.
- [24]. H. A. Basit, and S. Jarzabek, "A data mining approach for detecting higher-level clones in software," *IEEE Transactions on Software Engineering*, pp. 497-514, 2009.
- [25]. Z. Li, S. Lu, S. Myagmar *et al.*, "CP-Miner: A tool for finding copy-paste and related bugs in operating system code," in *Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation*, 2004, pp. 20.
- [26]. A. Michail, "Data mining library reuse patterns using generalized association rules," in *Proceedings of 22nd International Conference on Software Engineering (ICSE'00)*, Limerick, Ireland, 2000, pp. 167-176.
- [27]. N. Sahavechaphan, and K. Claypool, "XSnippet: mining for sample code," *ACM SIGPLAN Notices*, vol. 41, no. 10, pp. 413-430, 2006.
- [28]. T. Xie, and J. Pei, "MAPO: Mining API usages from open source repositories," in *Proceedings of the international workshop on Mining software repositories*, 2006, pp. 54-57.
- [29]. T. Zimmermann, P. Weisgerber, S. Diehl *et al.*, "Mining version histories to guide software changes," *IEEE Transactions on Software Engineering*, 31(6), pp. 429-445, June 2005.
- [30]. A. E. Hassan, and R. C. Holt, "Predicting change propagation in software systems," in *Proceedings of the 20th IEEE International Conference on Software Maintenance*, 2004, pp. 284-293.