© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

IMPLEMENTION DATA MINING IN SOFTWARE ENGINEERING

Himangi
Assistant professor, Dept of Computer science and engineering
Guru Jambheswar university of science and technology Hisar

Abstract: When it comes to software development, software companies generate enormous amounts of data. From the requirements phase all the way through to software maintenance, a new collection of data is generated at every step. Efforts are made to gather and keep data created in software repositories in order to improve the quality of the software. Software repositories include a vast amount of data, which is mined using different Data Mining methods to identify new patterns or highlights in the data. Study in this field has recently been a favourite multidisciplinary research subject for Software Engineering and Data Mining researchers. An attempt is made in this study to look at the numerous applications of data mining in software engineering, the various forms of software engineering data that may be mined, as well as different data mining techniques that are accessible and have been utilised by researchers to tackle their relevant issues. It's now time to narrow down which software engineering topic is grabbing academics' interest the most, based on this categorization.

Keywords: Software Engineering, Data Mining, Hidden Patterns, Software Design.

[1] Introduction

Academic scholars have been captivated by the topic of software engineering over the last few decades. Although the importance of software engineering has decreased over time, the pressure on software developers to produce high-quality products has risen in today's competitive world, where software has become an integral part of our lives. So many aspects of software engineering are being studied by scholars and practitioners, such as software quality, metrics and configuration management.

During the last several years, academics and practitioners have been shifting their attention to interdisciplinary study fields, and the most prominent area in this context is the combination of data mining and software engineering. It has the potential to be very beneficial in the quest for new and important information buried inside a mountain of accumulated data. It's a method for uncovering patterns in large amounts of data spread over a variety of repositories, including operational, distributed, external, and hypermedia databases. Researchers may discover new patterns in data contained in databases using a variety of Data Mining approaches. Organizations benefit from this data extraction since it aids in the process of making business choices.

There is a lot of data generated during software development that may be mined using data mining methods in order to enhance the quality of the programme being developed, in addition to traditional databases.

SE Data, which is gathered at different phases of software development and saved in repositories, is very valuable.

1.1 Software Engineering

Systematic software development is the focus of the discipline of software engineering. Those who design, build, maintain, test, and review computer software are known as software engineers.

Software engineering is an engineering method to the construction of a software system for systematic.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed

DOI : 10.5281/zenodo.7021762

Those who design, build, maintain, test, and review computer software are known as software engineers. As a synonym, "programmer" is sometimes used, although it may also lack implications of technical education and abilities.

The software development process is informed by engineering methodologies, which include the definition, implementation, evaluation, measurement, management, modification, and improvement of the software life cycle process itself. In order to ensure the integrity and traceability of the configuration and code throughout the system's life cycle, it largely relies on software configuration management. Software versioning is a common practise in the modern workplace.

BUGS
DOWNLOAD
APPLICATION
INSTALL
SOFTWARE ENGINEERING
LANGUAGE
NETWORK
SCRIPT
UPDATE
ComputerHope.com

Fig 1: software engineering

https://www.computerhope.com/jargon/s/software-engineering.jpg

1.2 Software design

Defining the architecture, components, interfaces, and other properties of a system or component is the focus of software design. Software architecture is another term for this. There are three stages in the software design process. Designing an interface, an architectural structure, and a comprehensive design are all parts of the process. The design of an interface is concerned with how a system interacts with its surrounding environment. Because of how complex the system's internal operations are, this takes place at a high degree of abstraction as well. The fundamental components of a system and the roles, attributes, interfaces, and interactions that occur between them are all part of architectural design. The internal parts of all key system components, including their qualities, relationships, and processing, as well as their algorithms and data structures, are all part of a detailed design., and other aspects.

Unit and integration tests as well as debugging are part of software development's fundamental activity: programming. During this stage, testing is often carried out by the programmer to ensure that what has just been produced is correct and to determine whether the code is ready to be transferred to the next stage.

With various methodologies, such as unit tests and integration tests, software tests are undertaken to provide stakeholders with information about the quality of the product or service being tested. It's a part of the overall quality of the software. It is usually done by a different developer or member of the quality assurance team than the one who authored the code as a distinct stage in the software development process.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

The operations necessary to offer cost-effective support for a software product after it has been shipped are referred to as software maintenance. Modifying and upgrading software programs after they have been distributed in order to fix bugs and enhance performance is known as software maintenance. A lot of software depends on the actual world, and when the real-world changes, so does the need for software maintenance. Removing and archiving obsolete and inactive components is an important part of software upkeep, as is fixing bugs and making improvements to existing components that have already been implemented. Therefore, by concentrating on maintenance, project costs can be kept down to a minimum of 40 percent to 80 percent.

Designing Software with the Following Goals in Mind

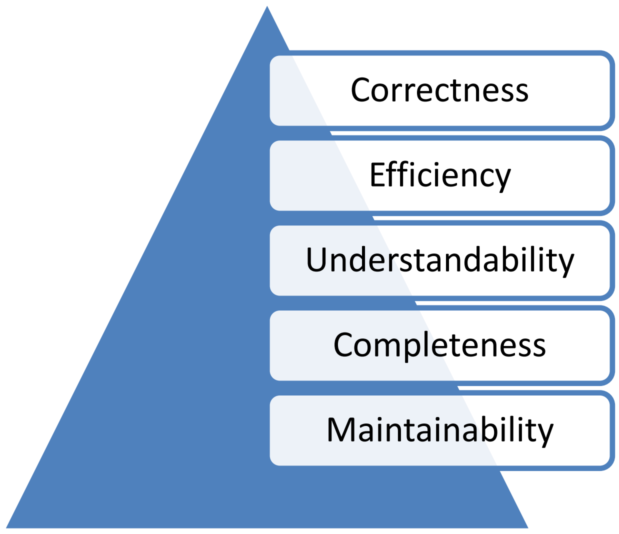

Fig 2: Goals of designing Software

Correctness: Proper implementation of all of the features of the system is a must for a good design.

Efficiency: A good software design should take into account the optimization of resources, time, and cost.

Understandability: A well-thought-out design should be simple to comprehend, which necessitates the use of modules and the layering of those modules.

Completeness: Everything from data structures to modules to external interfaces should be included in the design.

Maintainability: It is important that a customer's request for a modification is easily accommodated in the design of the program.

1.3 Data Mining

Finding anomalies, trends, and correlations in huge data sets to anticipate future events is known as data mining. You may utilize this information to boost income, save expenses, improve customer relations, and reduce risk and more by employing a wide range of approaches.

Large datasets are mined for patterns using approaches that combine machine learning, statistics, and database management systems. Extracting useful information from a dataset is the primary purpose of data mining, an inter-

DOI : 10.5281/zenodo.7021762

disciplinary discipline of computer science that aims to do just that. The analytical stage of "knowledge discovery in databases" is called data mining. Database and data administration, as well as interestingness metrics and complexity concerns and live updating are all part of the process.

Data mining is a misnomer since the aim is not the extraction of data itself, but rather the extraction of patterns and information from enormous volumes of data. Artificial Intelligence and business intelligence are two of the most commonly used buzzwords for large-scale data processing and computer decision support systems. As a marketing gimmick, the word "data mining" has been added to the title of Data mining: Practical machine learning with Java. Rather than using the specific phrases "data analysis" or "analytics," other relevant alternatives include "artificial intelligence" and "machine learning."

Steps of data mining process

The process of data mining involve following steps:

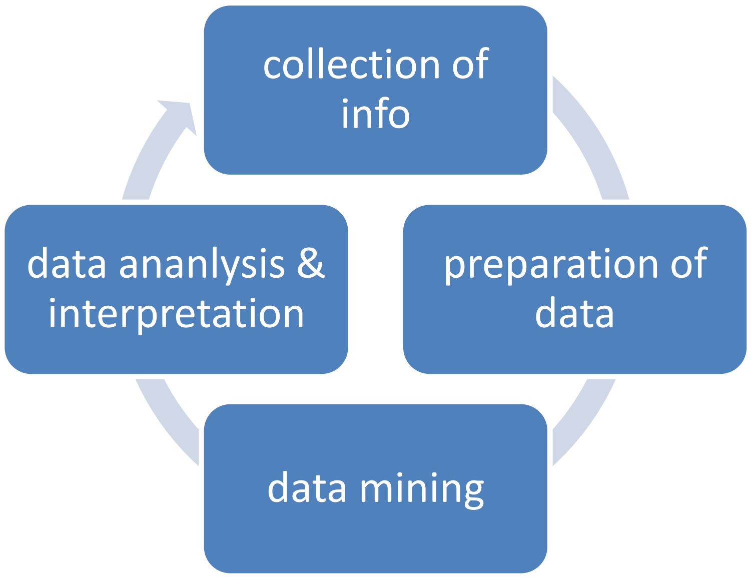

Fig 3: Data mining process

The collection of information. The process of locating and compiling data that may be used in an analytics program. An increasing number of big-data settings are using a data lake to store structured and unstructured data from a variety of different sources. It is also possible to use data gathered from outside the company. A data scientist often moves the data to a data lake for the rest of the process, regardless of where it originates from.

Preparation of data. It is at this point that the data is prepared for data mining. In order to repair mistakes and other issues with data quality, data cleaning is performed after the previous steps of data exploration, profiling, and pre-processing have been completed. Unless a data scientist is trying to study raw data for a specific application, data transformation is done to ensure that data sets are consistent.

The data mining. Data scientists first prepare the data before selecting the best data mining approach and implementing one or more algorithms to carry out the mining process. The algorithms used in machine learning applications are often trained on small samples of data before being tested on the entire dataset.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

Interpretation of collected data. Analytical models based on the data mining findings aid in commercial decision-making and other activities. An important part of the data scientist's job is to convey results to corporate leaders and end-users using visualisation and narrative strategies.

1.4 Hidden Patterns

Cleaning the raw data and analysing it with the help of a data mining programme is the first step in the process of discovering hidden patterns or trends. Any results that appear significant should be confirmed using typical statistical methods at this point.

For the most part, the data we gather is done so that we may look for trends or connections between different sets of data. It's possible to discern that pattern in a basic tabular representation of the data, depending on how the data and patterns are laid up.

- The "hidden pattern" issue, also known as the "sequence comparison" problem, entails finding a certain subsequence inside a text (rather than a string understood as a sequence of consecutive symbols).
- Analyzing data to uncover previously unknown correlations
- Anything that can be recorded or measured is referred to as DATA.
- When data points are clustered, those within one cluster are more comparable to one another than the data points inside other clusters.
- An array of values with several dimensions, often known as a DATA CUBE.

[2] Literature Review

This section shed light on past studies undertaken on Software Engineering, Data Mining, Hidden Patterns, Software Design. These are discussed ahead:

During the course of a software project's life cycle, it creates a great deal of data, both organised and unstructured, which may be used for many purposes. Various data mining methods may be used to extract all of these distinct kinds of information. Documentation, software configuration management data, source code, generated code and execution traces, issue tracking and bug databases, and mailing lists are the most essential data sources for software engineering, according to M. Halkidi et al [1]. There are three types of SE data repositories: historical repositories, run time repositories, and code repositories, according to A.E. Hassan et al [2]. There are many additional key data sources that can be extracted from CVS logs as well as other data repositories such as Git, GitHub or Junit that can be used for mining. [3] Chaturvedi and his colleagues discussed these data sources in their research work.Information retrieval was the focus of a method presented by Huffman et al. [4]. Their approach was to improve the extraction of high and low level requirements by treating the documents' universe as a union of design elements and individual requirements and to map the problem of requirements tracing into finding the similarity between the vector space representations of high and low level requirements, reducing it to an IR task in the process. Additional to this study was done by Ankit Dhamija and Sunil Sikka in [5] in order to uncover factors that influence how software engineers use data mining technologies to achieve outcomes.

A programme called SoftChange was suggested by German and colleagues [6] that utilises textual data from open source software and conducts data validation on it, which performs retrieval, summarization and validation operations.Open source software was also the focus of Jensen et al [7], who suggested a method for identifying software processes. Entity resolution and social network analysis were both used in the process of text extraction.

Using the source code repository's modification history, Williams & Hollingsworth [8] developed a new way for enhancing static analysis methods used to discover defects.Association rule extraction approaches were used to evaluate defect data by Morisaki et al [9]. Bugs, specifications, and design modifications are all examples of software

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

faults. Extraction of important information and rules related with defect rectification efforts was accomplished via the use of an extended association rule mining approach based on data collected from defects.

The graph mining methods provided by Chang et al [10] may be used to locate conditional rules in a code base and to extract rule violations that highlight overlooked conditions that need to be addressed. [11] In order to uncover conditional rules, dependency graphs were employed and frequent item set mining and frequent sub graph mining approaches were applied.

As described by Dickinson et al. in [11], a clustering technique-based approach was presented for filtering profiles based on comparable traits in order to select executions for conformance testing.The remaining clusters are then used to determine execution profiles.Using an infofuzzy network (IFN) with a tree-like topology, last et al[12] propose to automate the processing of input and output data.

Using Bowring et al's [13] method, a set of programmes' executions are analysed and classifiers of software behaviour are developed. A Markov model was utilised to encode the project execution profiles. After that, an agglomerative clustering technique is used to group the Markov models of specific programme runs.

A mining technique developed by Liu et al [14] employed a statistical approach to assist programmers in manual debugging.

Liu et al [15] suggested a new approach for analysing logical errors based on data mining. Using closed graph mining and support vector machines to classify programme executions, they came up with this solution.

Kannelopoulos et al [16] proposed a clustering-based strategy for acquiring information from source code in order to understand an object-oriented system and assess its maintainability.

Citation	Author/year	Objectives	Methodology	Limitation
[1]	D. Spinellis/2011	Data mining in software engineering	Data mining	Lacks accuracy
[2]	R. C. Holt/2004	Predicting change propagation in software systems	Predicting, software systems	Less performance
[3]	Singh V.B/2013	Tools in Mining Software Repositories	Data Mining	Less focus on efficiency
[4]	Huffman/2003	Improving requirements tracing via information retrieval	Information retrieval	Focus only on statistical tests
[5]	Sundaram/2005	Analyst feedback based results	Text mining	Less efficiency
[6]	D. German/2003	Open-source projects	Automation on data mining	Less accurate
[7]	C. Jensen/2004	Software process discovery	Data mining	Narrow focus
[8]	Williams/2005	Automation on data mining	Data mining	Less performance
[9]	Morisaki/2007	Defect data analysis	Data analysis	Less focus on efficiency
[10]	Chang /2008	Tracing neglected conditions in software	Data mining	Less real life application

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

Citation	Software Engineering	Data Mining	Hidden Patterns	Software Design
[1]	Yes	Yes	No	No
[2]	No	No	No	Yes
[3]	No	Yes	No	No
[4]	Yes	No	No	No
[5]	No	Yes	No	Yes
[6]	Yes	No	No	No
[7]	No	Yes	No	Yes
[8]	Yes	No	No	No
[9]	No	Yes	No	Yes
[10]	Yes	Yes	No	No
[11]	Yes	No	No	No
[12]	No	Yes	No	No

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

[3] Problem Statement

However, the authors here believe 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. Software engineering and data mining are two closely related fields that are discussed in this review article. He or she seeks to find out what kinds of data may be mined and which tools are ideal for doing so, as well as how the various techniques can be applied to different types of software engineering data. Finally, the author makes an attempt to determine the most significant SE region based on this categorization.

[4] Need of Research

To give a three-way analysis of diverse Software Engineering Data, the researchers in this study have classified the various kinds of tools and provided a summary analysis of various Data Mining Tools and their emphasis area in Software Engineering data sources.. According to the extensive literature review and tools analysis, most of the work and most of the tools proposed by researchers until today have been on software code, and thus there is a great deal 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 associated with Software Engineering in spotting and solving problems in the software.

[5] Scope of Research

It is more common for scientists to utilize pre-existing tools to do software engineering data mining activities than it is for them to design their own custom software tools. These technologies were used by researchers to extract data from repositories, detect patterns, learn, and forecast.

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 andW. Scacchi, Datamining for software process discovery in open source software development communities. In Proceedings of International Workshop on Mining Software Repositories (MSR), 2004.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

[8]. C.C.Williams and J.K.Hollingsworth,Automatingmining of source code repositories to improve bug finding techniques, IEEE Transactions on Software Engineering 31(6) (2005), 466–480.

[9]. S.Morisaki,A.Monden andT.Matsumura, Defect data analysis based on extended association rulemining. InProceedings 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 Dimining Approach to Automated Software Testing, In Proceeding of the SIGKDD Conference, 2005.

[13]. J. Bowring, J. Rehg and M.J. Harrold, Acive 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 ow 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 noncrasinh 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 tokenbased 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 higherlevel clones in software," IEEE Transactions on Software Engineering, pp. 497-514, 2009.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 13 Issue: 04 | July - September 2022
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
DOI : 10.5281/zenodo.7021762

[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.