Corrosion Management ISSN: 1355-5243
(https://corrosion-management.com/)
Volume 34, Issue 01 – June 2024
SCOPUS

INTEGRATION OF DATA MINING IN ADVANCED SOFTWARE ENGINEERING PRACTICES

Manushi[1] (Research Scholar)
Dr. K.P Yadav[2] (Research Supervisor)
Department of Computer Science
[1,2] Sikkim Professional University, Gangtok, (Sikkim)

Abstract

Knowledge discovery and data mining have been incredibly useful in many fields, including manufacturing, medicine, and administration. Their usefulness also includes dealing with difficult aspects of software engineering, such security and adaptability. One of the most important roles of analysts in software engineering is to glean information from many sources, including user assertions. Data mining offers a variety of tools and algorithms for effective data collection. In order to improve data collecting and analyst decision-making, this study aims to integrate software engineering methodologies with data mining algorithms.

Keywords: *Software Engineering, Data Mining, Integration, Advancement*

Introduction

Data mining, often called knowledge discovery in databases, is the process of discovering useful, hidden information in databases by analysing existing data. Even though the terms are often used interchangeably, data mining is really an essential part of knowledge discovery when it comes to databases **(Ohira et al., 2015).**

- Knowledge Prediction in Databases is an iterative process that starts with collecting raw data and ends with new knowledge:
- Data Cleaning: In this step, any data that is useless or too noisy is removed from the dataset.
- Data Integration: In this step, many data sources, which may be different from one another, are combined into one.
- Data Selection: Finding and retrieving analysis-relevant data from the collection.
- Data Transformation: This step involves converting the chosen data into formats that are appropriate for mining.
- One crucial phase is data mining, which involves applying clever approaches to uncover patterns that could be valuable.

- Pattern Evaluation: By following certain algorithmic techniques, intriguing patterns that represent knowledge can be recognised.
- Knowledge Representation: In the last step, visualisation techniques are used to visually portray the obtained knowledge to the user. This helps with understanding and interpreting the data mining results.

To create a data warehouse, for instance, the pre-processing phase can incorporate data integration and data cleaning. Data consolidation occurs as a result of either data selection or data modification; this is illustrated by data warehouses **(McCormick & Salcedo, 2017).**

Software engineering is an approach that focuses on efficient development, high-quality, affordable, and maintainable software through design, implementation, and refinement. Analysis, design, evaluation, implementation, testing, maintenance, and reengineering are all steps in the software development life cycle that are followed methodically **(Yethiraj, 2012).**

Planning, collecting requirements from stakeholders, and designing and deploying the development environment are the three key steps that must be taken before software development can begin. Once these are set up, the project moves on by means of a succession of development initiatives that build upon one another.

Important steps in creating software include:

- The first step in starting a project is figuring out what the client needs, forming the development team, and creating a plan.
- Perspire Planning: Participants in a team may work on multiple tasks at once and in any sequence they see fit.
- Developing the Timetable: Specifying the Required Features and Major Threats.
- Team Building: Assembling Groups for Maximum Efficiency in Workflow.
- Reuse Planning: Finding ways to reuse things and figuring out how much it will cost.
- Finding issues with the project, determining their severity and probability, and calculating the risks involved are all part of risk reduction planning.
- Articulating Logical Architecture: Outlining the framework and recurring patterns of an undertaking.
- Finding risks, measuring issues, and doing safety analyses as part of an initial safety and reliability examination.

Software analysts are vital in collecting data for these procedures. Collecting data from a wide variety of sources, including presentations, PDFs, and papers, may be a time-consuming ordeal. The use of automated approaches has increased in recent years due to the fact that manual search methods often result in errors.

Data mining for SE

There has been a lot of success in mining software engineering data as a research direction in recent years, both in academia and industry. The authors of this study believe that Software Intelligence (SI) will soon replace traditional data mining methods in the field of software engineering. Although SI is still in the future, there have been encouraging developments in the area of Mining Software Repositories that point to its possible realisation in the not-too-distant future. The concept of SI is becoming increasingly important in software engineering research as it aims to continue making a difference in modern software practice.

According to Lozano et al. (2010), this position paper surveys the present and future of SI research and practice and lays out plans for future studies to mine data from software engineering in a way that makes SI a reality.

An emerging field of study, text mining applies data mining methods to the problem of information overload. But because text mining is all about unstructured data, preparing document collections is a big part of it. This includes a variety of methods for analysing intermediate representations as well as storing them. Text mining systems sometimes start with unlabeled collections of documents and automatically sort them into categories based on phrases, relationships, or other entities pulled straight from the documents (Jo, 2019). A variety of data mining operations can be performed on the documents using the extracted categories and relationships. In order to facilitate software analysts' work, this article suggests incorporating a novel data mining strategy into software engineering.

As a hybrid strategy combining data mining clustering methods with software engineering analysis phases, the probabilistic clustering method will be utilised in this research. Data is seen as a random sample from various probability distributions, with each distribution representing a different user aim, in this approach.

With probabilistic clustering, picking a data point at random from a corresponding distribution is the central idea. Each distribution's natural cluster is the region surrounding its mean;

characteristics like variance and mean are connected with each cluster. To facilitate the estimation of assignment probabilities, each data point is equipped with attributes and a cluster ID. The underlying assumption is that each point is believed to belong to a single cluster.

As an objective function, the customer's goal guides the two-step iterative optimisation process known as Expectation Maximisation (E-M). By estimating probability in the Expectation (E) stage—which is similar to soft reassignment—and by maximising estimation findings until convergence, the Maximisation (M) step determines an approximation to the customer target given current soft assignments.

To speed up EM iterations and help locate better local optima, tricks like decision trees, KD-trees, and specific data indexes are useful. Probabilistic clustering has many useful features, such as being able to adapt to complicated user goals, stopping and starting with successive batches of data, being able to assign tasks online at any iterative stage, and having cluster systems that are easy to understand because of their clear probabilistic basis. Since overfitting can occur when there are too many parameters, the Bayesian framework makes it easier to determine the appropriate number of clusters (k) from a data mining standpoint.

Probabilistic clustering, with its interpretability, adaptability, and efficient convergence qualities, is a strong tool for analysing software engineering data.

Result

We have taken set of sample user objectives of the software engineering for analysis purpose. We tested performance effectiveness with the human analyst with our probabilistic clustering method. Below diagram shows the effectiveness of the probabilistic clustering method. The result value is given out of ten.

Table 1.1 Summary of the results

	Analyst	Proposed Method
Missed Requirements	4	2
Accuracy	6	9
Remembrance	4	8
Trace Back	3	9
Final Result	6	9

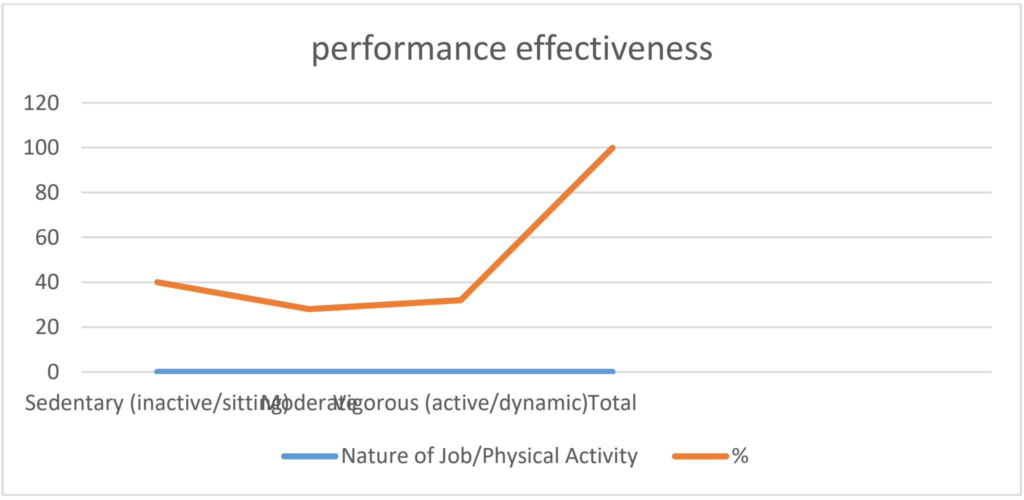

Figure 4.10. Performance effectiveness

The comparative analysis between the human analyst and the proposed probabilistic clustering method for analyzing a set of sample user objectives in software engineering reveals notable differences in performance across various metrics. The results, scored out of ten, indicate: The proposed method consistently outperforms the human analyst, demonstrating reduced instances of missed requirements, higher accuracy, enhanced remembrance, and superior trace-back capabilities. The overall final result also reflects the effectiveness of the probabilistic clustering method, scoring notably higher than the human analyst. These findings suggest that the proposed method holds promise in improving the accuracy and reliability of software engineering analysis compared to traditional human-centric approaches

Conclusion

The software business has benefited greatly from data mining and software engineering developments during the last decade. To improve analysts' analytical skills, our research has combined data mining with software engineering techniques. Improved accuracy and less room for human mistake are two benefits of this method over traditional human analysis.

By developing more effective and efficient analysis approaches, we think our work will substantially help the software engineering community. Clustering analysis of user objectives was carried out using the probabilistic clustering method in our research. There are a plethora of clustering methods accessible in data mining, although probabilistic clustering is one of the most powerful.

Our long-term goal is to improve our analysis and draw stronger findings by investigating and using different clustering algorithms. We can find out which algorithm works best for software engineering applications by comparing them. The area of software engineering will benefit from this investigation, and software development processes will become more efficient and of higher quality as a whole.

ISSN: 1355-5243

Reference

Ohira, M., Kashiwa, Y., Yamatani, Y., Yoshiyuki, H., Maeda, Y., Limsettho, N., ... & Matsumoto, K. (2015, May). A dataset of high impact bugs: Manually-classified issue reports. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories* (pp. 518-521). IEEE.

McCormick, K., & Salcedo, J. (2017). *IBM SPSS Modeler essentials: Effective techniques for building powerful data mining and predictive analytics solutions*. Packt Publishing Ltd.

Lozano, A., Kellens, A., Mens, K., & Arevalo, G. (2010, October). Mining source code for structural regularities. In *2010 17th Working Conference on Reverse Engineering* (pp. 22-31). IEEE.

Yethiraj, N. G. (2012). Improvement of Software Maintenance and Reliability using data mining techniques. *International Journal of Data Mining Techniques and Applications*, *1*(2), 101-6.

Jo, T. (2019). *Text mining* (Vol. 45). Cham, Switzerland: Springer International Publishing.