Advances in Artificial Intelligence and Machine Learning; Research 6 (1) 4992-5002 Received 20-11-2025; Accepted 05-02-2026; Published online 12-02-2026

Data-Driven Optimization of Bug-to-Expert Assignment in Software Development

Racharla Mamatha mamatha.racharla@gmail.com
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation,
Hyderabad -500075, Telangana, India.

P Lalitha Surya Kumari vlalithanagesh@gmail.com
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation,
Hyderabad -500075, Telangana, India

Adepu Sharada sharada@gnits.ac.in
Department of CSE,
G. Narayanamma Institute of Technology & Science
Hyderabad-500104. Telangana, India.

Corresponding Author: P Lalitha Surya Kumari

Abstract

Purpose: The aim of this study is to use the data of past projects and align the skills of employees with the needs of certain bugs, thus enhancing the quality of outputs and reducing time taken to find solutions to a specific issue. To build a universal, competency-based strategy of bug assignment, the research also tends to create a framework, which could be extended to other projects.

Methodology: The data of one of the privately held companies that are under analysis includes important project items such as Bug ID, Product, Component, Assignee, Priority, Severity, and Skill Level. These are essential aspects that are analyzed to identify trends that can lead to a smarter allocation process. Data-driven allocation strategies are created with the help of pattern mining rules and machine-learning algorithms used to disclose the hidden correlations between characteristics. Its methodology is directed at the aligning of personnel talents and the difficulty of the task within the complex decision-making environment based on the analysis of previous information in the software development projects.

Results: The findings indicate that the algorithm is capable of identifying good solutions when it is used to combine diverse tasks, including project history analysis, determining the severity of bugs and deploying relevant skills in debugging them. The accuracy achieved is 97.02%

Conclusion: This study provides a new method of project efficiency and maintenance optimization through the ability to match employee skills with the needs of the tasks with the help of machine learning algorithms and historical data. When applied in a large number of projects, the proposed method can transform the process of managing and allocating tasks in the sphere of software development and other industries. The framework provides a policy of enhancing efficiency and management of resources in various industries than software engineering. On the whole, when businesses aim at enhancing the productivity, the report suggests the smart approach to resource distribution and the simplified procedures that can be scaled.

Keywords: Software development, Frequent pattern mining, Rules for association, Allocation, Optimization.

1. INTRODUCTION

The process of discovering helpful information that is usually buried in large volumes of data or data repositories, and this are called data mining [4], which can greatly transform the world. Effective talent recruitment and building of effective strategies of large organizations and government bodies require the understanding of the competency and the nature of the employees. Using data mining technologies, the organizations will be able to retrieve the human resource information in databases and give a detailed information about the employee performance[1, 6, 7], identification of patterns, and identification of the missed factors that can influence decisions. There are many industries in which data mining methods have been successfully used in finance, healthcare, retail, manufacturing, telecommunications, and customer service. These techniques have been applied in Human Resource Management (HRM) in intelligent defect diagnosis, talent acquisition, categorization of employees, evaluation of performance, and planning strategies. The data mining techniques employed by researchers have been varied and some of them include artificial neural networks, K-means clustering, decision trees, predictive modeling and association rule mining using the Apriori algorithm. These practices have been incorporated in HR systems to enhance recruitment procedures, performance appraisals, pay scales as well as employee welfare schemes using data mining & Fuzzy logic techniques[11, 12]. Application of data mining to the HR practices like hiring, personnel selection, training and job performance assessment is promising to a large extent. However, there are still challenges of employee turnover and retention. Specific measures should be taken to overcome these difficulties with the help of building a robust corporate culture, laying stress on talent forecasting and planning, and developing talent market development programs. Human resource management should be given priority in the organizations because it directly influences the forces within the workplace and the competitive advantage. As an example, project management, including budgets and project schedules, is crucial in software development companies, and fair and effective distribution of tasks is a significant factor. The assignment of the tasks may reduce the productivity of the teams, and the quality of the projects may be compromised when the allocation of tasks is incorrect or ineffective. Proper allocation of human resource to projects can result into increase in revenue, satisfaction of clients and the general success of the organization.

HRM assignment of employees to project teams is a complicated task that requires due consideration of a number of factors. Project managers should be able to predict the labour and skills needed to bring successful projects. Poor allocation strategies may hinder developments and counter expected gains. An accurate process of selection is important so that members of the project can be allocated according to their talent and what is being required by the jobs. With the increase in the complexity of the software projects, the necessity of efficient human resource allocation methods becomes more essential. The technological developments are making HR allocation procedures easier[14, 17], making tasks management more efficient and burdening the project managers less. Having a clear HR allocation plan[8] makes the planning and schedule planning simple as resources are utilized at the best level possible to meet the objectives proposed.

2. BACKGROUND STUDY

Leif Jonsson [1] and his co-authors highlight the need to properly assign bug reports when it comes to the maintenance of software, particularly large development projects. Proper assignment helps to avoid costly errors. Despite the existing research on the application of machine learning to bug allocation in open-source projects, the application to enterprise settings is surprisingly scarce. Their analysis shows that machine learning classification methods can be used to enhance the performance of bug assignments in the industrial context significantly, especially with ensemble learning methods such as Stackable Generalization (SG).

Junjie Che[5] and colleagues analysed the huge online service systems at Microsoft and discovered that the allocation of incident report is wrong especially the critical ones that are classified as high severity leading to serious time and financial consequences. Between 4.11 per cent and 91.58 per cent of reports require reassignment, and erroneous assignments can add up to 10.16 times to incident triage. The existing methods used to triage bugs can be modified to meet the requirements of incident triage, however it is necessary to make significant improvements to deal with early allocations. This practical study that particularly focuses on industrial use is informative to researchers and practitioners who would wish to advance the incident triage methodology in large web-based systems.

Anjali Goyal and Neetu Sardana[3] also point at the critical role that effective bug allocation assumes in dealing with a large number of problem reports during the software development process. In their studies, they found that information retrieval methods are more useful in the context of assigning bugs to individuals compared to machine learning because the former outperforms machine learning when it comes to suggesting developers to work on particular problems. This conclusion is supported by the analysis of repositories of such famous projects as Mozilla, Eclipse, GNOME, Open Office.

Ramya C., Paramesh S.P., and Shreedhara K.S[2]. suggest that automated classification through the standard machine learning technique can be used as a solution to the issue of improper routing in IT service desk tickets. Their study evaluates the accuracy of three group techniques, which are Bagging, Boosting, and Voting in enhancing categorization accuracy. Ensemble classifiers performed better than their respective base counterparts in experiments done using real-world data of a large-scale corporate IT system. The results mentioned above show many benefits, such as a more user-friendly interface, quicker fixing of issues, greater productivity, increased customer satisfaction, and general improvement in the business.

3. METHODOLOGY

The research is based on a database obtained in one of the Indian privately owned businesses, where 150 employees work. The study is aimed at examining the different characteristics associated with projects in order to unearth trends, which are:

- Bug ID: Each bug has a unique code assigned to it.
- Product: refers to the type of software product related to the bug.
- Component: Identifies the exact location within the software of the bug.
- Assignee Name: This name is assigned to the individual to repair the bug.
- Priority: This is the urgency of the bug fix according to its importance.
- Severity: Assesses the degree of effects of the bug, the problems may be the trivial ones or the issues that are critical.
- Skill: defines the specific abilities of the person that will be tasked with handling the bug.

The aim is to determine trends, as well as correlation, that can enhance practices relating to bug assignment. Through the analysis of the data provided by the company, this study will help to refreeze the matching of the bugs with the employees with references to the skills fit and previous information. The data includes the records that are associated with the bug assignments that are managed by the software development team. The standard pattern mining tools are used to identify the patterns in assigning various types of bugs to various team members. The level of experience of the workers, the categorization of bugs, the ratings on severity and the time of task are some important factors to take into account.

The results are expected to provide meaningful information that will improve the effectiveness of the team and the practice of a bug assignment. The identification of these trends can result in the achievement of better working results, the streamlining of the work process, and a more efficient matching of bugs to the competencies of the team members. The researchers suggest a multi-step algorithmic approach to bugs optimization as shown in FIGURE 1:

Step 1: Obtaining a wide range of data of the private firm.

Step 2: Use data cleansing and preprocessing techniques to prepare the obtained data to be used in training machine learning models. Adopt the use of optimization methods to find and mine the prominent trends in the data.

Step 3: Develop a system to match the proficiency of each employee to the seriousness of the bugs.
This strategic fit aims at maximizing resource utilization by ensuring that the best people are assigned to deal with specific levels of bugs.

Stage 4: Compare and in-depth analyze the various machine learning approaches. In order to determine the effectiveness of each approach, determine their effectiveness based on significant measurements.

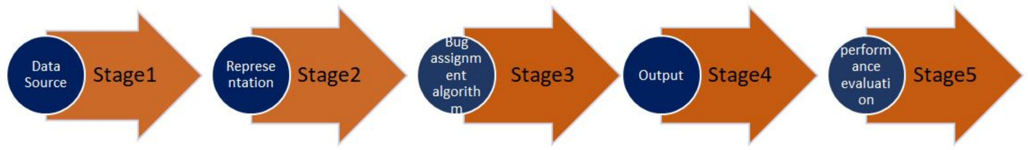


Figure 1: Workflow diagram of bug assignment process

Stage 5: Compare the best bug assignment optimization algorithm based on the data of accuracy and performance as shown in FIGURE 2.

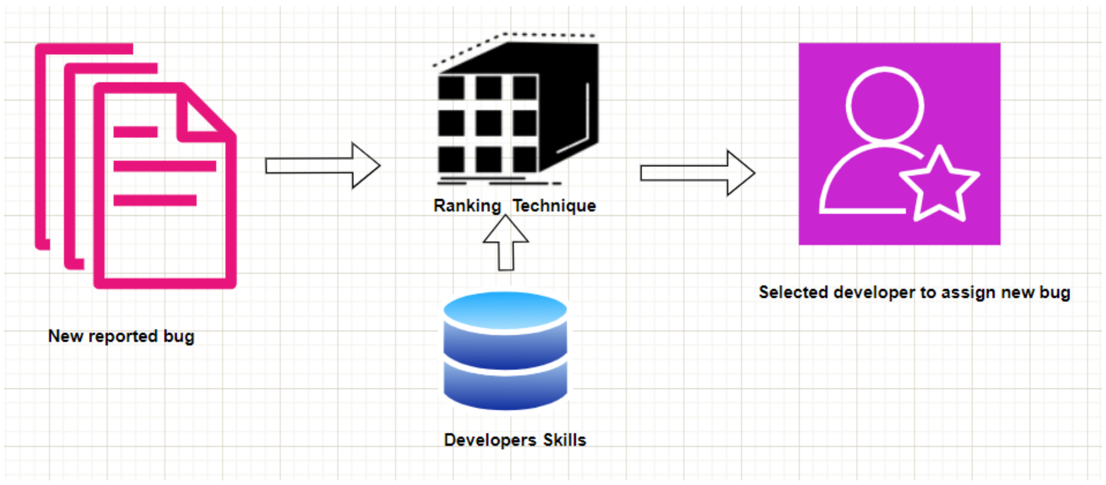


Figure 2: Bug to Employee Assignment method

Software development has bug tracking or problem tracking systems that collect and manage bug information. The techniques are necessary in identifying, documenting, and managing the software bugs. Many of the popular issue tracking solutions have been heavily utilized during the life cycle of software development. These systems are typically found by testing professionals or quality managers and put the information about software errors into them. Bug tracking solutions enable stakeholders and project managers to access bug data fully to track and evaluate it and receive information about bugs. This paper concentrates on the attributes, which are presented below in the TABLE 1, of bugs, but most studies on software bug triaging use publicly available data on freely available software projects.

Six essential steps comprise the study's suggested methodology:

Step 1: Data Collection A collection of bug reports, each with a corresponding severity rating, and a list of staff members and their skill sets is the input data.

Step 2: Data Preprocessing To enable more efficient analysis, convert the bug severity and employee skills into categorical values (e.g., low, medium, high).

Step 3: Frequent Pattern Mining To find common itemsets associated with problem severity and employee skills, use frequent pattern mining methods like Apriori or FP-Growth.

Table 1: Characteristics of a software bug report.

S.No	Attribute	Meaning
1	Bug id	A bug id is a unique integer number
2	title	Brief summary of a reported bug
3	description	Detailed information of reported software bug
4	priority	It represents priority of the bug
5	severity	It represents severity of the bug
6	component	Used Software component in project
7	state	state of software bug
8	opened/submitted	first-time submission date of bug
9	Modified/updated	software bug last updated date
10	assigned-to	Developer to whom the bus is assigned
11	reported-by	Developer details who has identified and reported the bug

Step 4: Rule Generation for Resource Allocation Create association rules using the frequently occurring itemsets as a guide. These guidelines forecast the probability that workers possessing a particular skill set will deal with various patterns of bug severity.

Step 5: Resources Allocation.

We are going to evaluate the severity of any bug report and the related features in correspondence with the pre-developed criteria. Once we have a good fit in terms of necessary skills, we can outsource the task to the most able person or team to carry out the task.

Step 6: Introduction of the Resource Allocation.

Each bug report will show the assigned individual or team and a breakdown of his or her qualification. The approach enables us to organize resources allocation in a systematic, data-based way, which will guarantee the organization of the resources with regard to skills of our staff and the urgency of the problems.

3.1 Substantia Nigra Mining Frequent Patterns (MFP)

Mining of frequent patterns commonly known as MFP is a crucial data mining method whose aim is to identify the recurrent trends or item groupings among vast sets of data. This method is versatile and can be used in various areas such as web mining, DNA sequence analysis and market basket research. The main goal of MFP is to find groups or events which are observed within the dataset with a high frequency, so that organizations can make informed decisions with the help of data and use the knowledge in the area of product associations.

3.2 Process Flow of Frequent Patterns Mining of Bug Assignment:

The frequent pattern mining algorithm works on a number of processes to compose useful information of data in terms of transactions or sequence. We begin the process by preparing the data, in which we collect the dataset and carry out cleaning, which is solving any missing entries and any duplications. Next we convert the cleaned data into itemsets or sequences, which constitute item collections of transactional data and the sequences of sequential data. We then establish a support threshold in order to decide what frequency is needed to consider a pattern frequent. The data is then swept to assess the frequency of possible itemsets or sequences. Patterns that fail to meet the support criteria are removed and we rebuild the candidate pool to use in the next iteration as well as remove unused itemsets. The cycle repeats itself till we are left either with no new prevalent trends or with patterns of maximum length. In the end we give the corresponding support counts and the identified frequent itemsets or sequences. As a further step, we can consider association rule mining that will demonstrate relations between items which we will evaluate with the help of a simple level of confidence. The main rules will be offered with their confidence rates and measuring scales after the analysis of the created association rules on the basis of such metrics as lift, conviction, and interest.

3.3 Metrics Used in FPM

Frequent Pattern Mining (FPM) employs various metrics to evaluate the significance and quality of the discovered patterns. The most intriguing and significant patterns are filtered out with the aid of these measurements. Here are a few often used metrics:

Support: Support quantifies the frequency with which a specific pattern occurs within the dataset. It is determined by dividing the total amount of transactions by the number of occurrences that have the pattern. More frequent patterns are indicated by higher support levels.

Confidence: Confidence shows the conditional chance that the subsequent item or items will exist in a transaction given the presence of the antecedent item or items, indicating the dependability of an association rule. Stronger correlations are indicated by higher confidence levels.

Lift: In an association rule, lift measures how strongly the antecedent and consequent elements are related to one another. It contrasts the rule's actual support with what would be expected if the antecedent and consequent were unconnected. A positive correlation is indicated by a lift value more than 1, whilst a negative correlation is shown by values less than 1.

Conviction: Conviction quantifies how dependent the antecedent and subsequent things are on one another. It is computed as the ratio of the probability that the rule will be broken to the probability that the next item will not occur in a transaction. Stronger relationships are indicated by higher conviction levels, whilst weaker dependencies are suggested by values nearer 1

Interest: Interest evaluates an association rule's relevance by contrasting the observed support with what would be predicted if the antecedent and consequent were independent. A more intriguing and important regulation is indicated by a higher interest value. The most important measures of this paper to assess and quantify the quality of patterns that have been discovered are Support and Confidence. The methods are applicable in establishing the existence of vital and credible relationships in the data collection.

4. RESULTS AND DISCUSSION

Use of ROC curves is done so as to measure and estimate efficiency of various machine learning techniques. The curves would depict the distinction that a model would make between positive and negative set by plotting the true positive rate (sensitivity) and the false positive rate (1-specificity) at different levels of classification. Comparison of ROCs reveals that Support Vector Machine (SVM) gives the best desired results on the test data are shown in FIGURE 3 and FIGURE 4.

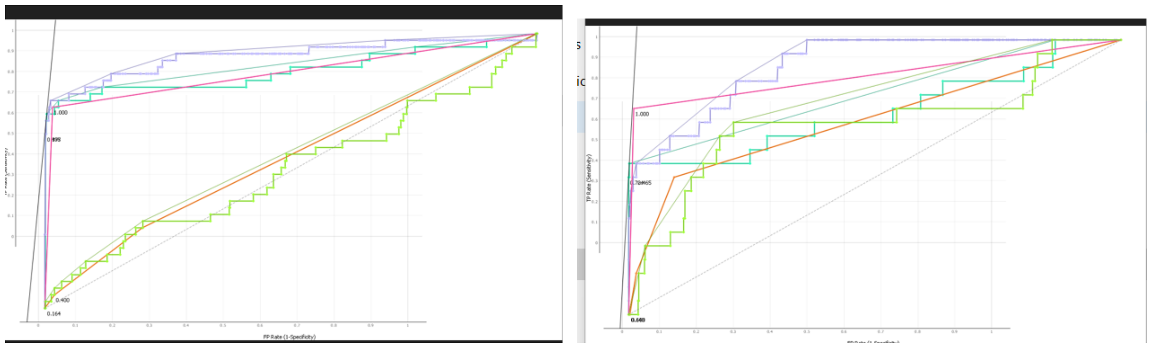


Figure 3: ROC curve showing minor and critical bug category

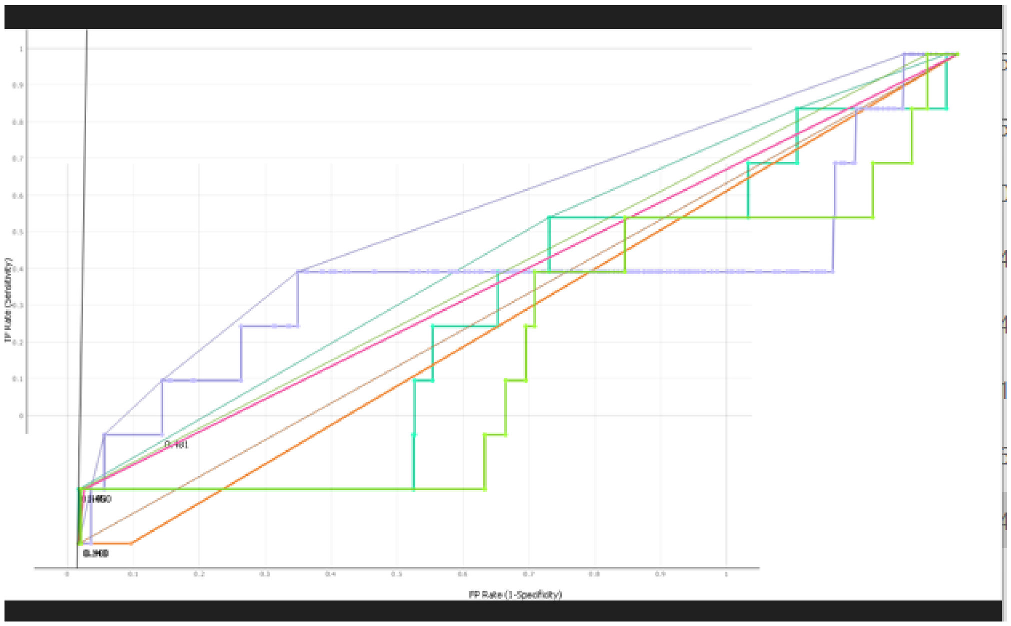


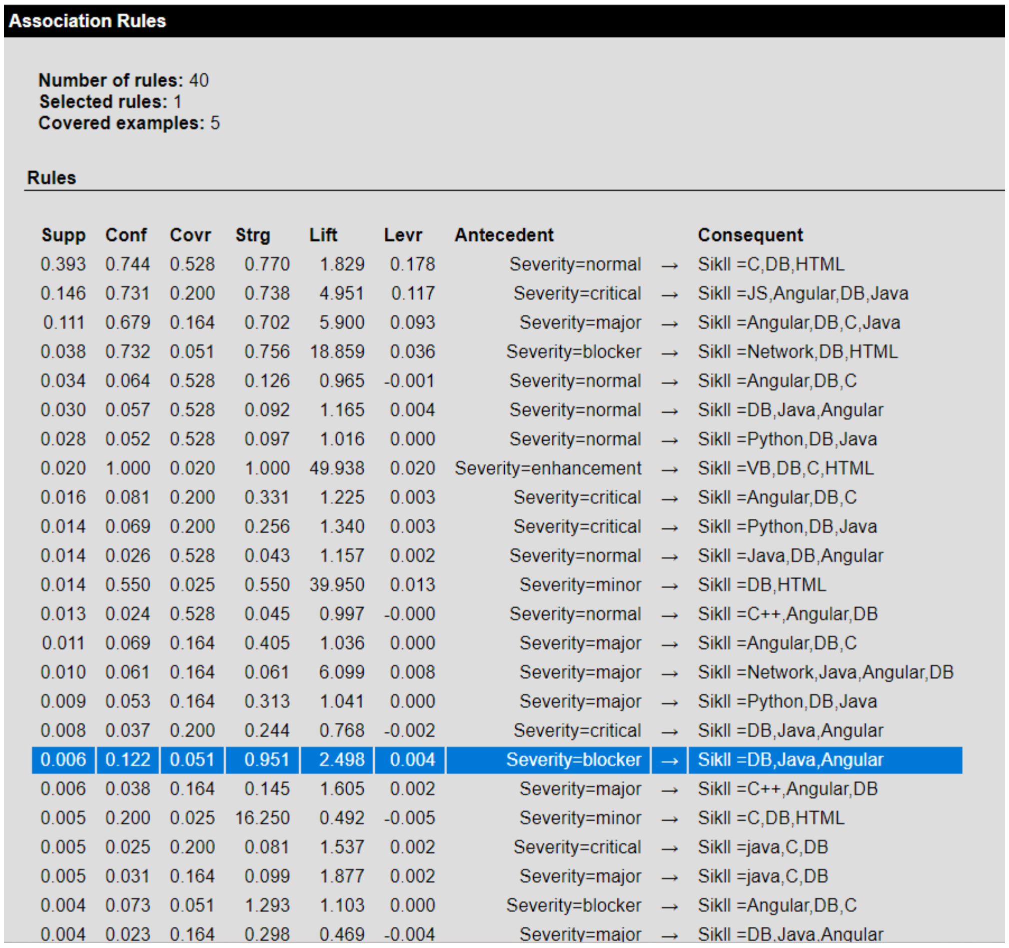
Figure 4: ROC curve for trivial bug category

The findings prove that the algorithm can find good solutions in the right way by a number of actions being combined which includes studying the history of the project, diagnosing the severity of the bugs, and providing the right skills to fix them as shown in TABLE 2. To some extent, this approach examines the frequency with which various patterns in the data have taken place.

Table 2: Bug assignment comparative study with existing study

S.No	Paper	Methodology used	Performance
1	Bhatttacharya et. al. 2012 [9]	Naïve Bayes and SVM	Accuracy 86.09%
2	Xuan et. al. 2014 [10]	Feature Selection and Instance	Accuracy 81.3%
3	Yin et. al. 2018 [15]	feature selection algorithm	Accuracy 71.2%
4	Xian et. al. 2016 [13]	Multi label learning	F-score = 0.56 – 0.62
5	Xi et. al. 2019 [16]	Classification	Accuracy 0.799%

4	Proposed study	ML with Mining algorithms	Accuracy 97.02%
			F-score = 90.05%
			Precision = 84.06%
			recall = 97.23%


Association Rules

Number of rules: 40
Selected rules: 1
Covered examples: 5

Rules

Supp	Conf	Covr	Strg	Lift	Levr	Antecedent		Consequent
0.393	0.744	0.528	0.770	1.829	0.178	Severity=normal	→	Sikll =C,DB,HTML
0.146	0.731	0.200	0.738	4.951	0.117	Severity=critical	→	Sikll =JS,Angular,DB,Java
0.111	0.679	0.164	0.702	5.900	0.093	Severity=major	→	Sikll =Angular,DB,C,Java
0.038	0.732	0.051	0.756	18.859	0.036	Severity=blocker	→	Sikll =Network,DB,HTML
0.034	0.064	0.528	0.126	0.965	-0.001	Severity=normal	→	Sikll =Angular,DB,C
0.030	0.057	0.528	0.092	1.165	0.004	Severity=normal	→	Sikll =DB,Java,Angular
0.028	0.052	0.528	0.097	1.016	0.000	Severity=normal	→	Sikll =Python,DB,Java
0.020	1.000	0.020	1.000	49.938	0.020	Severity=enhancement	→	Sikll =VB,DB,C,HTML
0.016	0.081	0.200	0.331	1.225	0.003	Severity=critical	→	Sikll =Angular,DB,C
0.014	0.069	0.200	0.256	1.340	0.003	Severity=critical	→	Sikll =Python,DB,Java
0.014	0.026	0.528	0.043	1.157	0.002	Severity=normal	→	Sikll =Java,DB,Angular
0.014	0.550	0.025	0.550	39.950	0.013	Severity=minor	→	Sikll =DB,HTML
0.013	0.024	0.528	0.045	0.997	-0.000	Severity=normal	→	Sikll =C++,Angular,DB
0.011	0.069	0.164	0.405	1.036	0.000	Severity=major	→	Sikll =Angular,DB,C
0.010	0.061	0.164	0.061	6.099	0.008	Severity=major	→	Sikll =Network,Java,Angular,DB
0.009	0.053	0.164	0.313	1.041	0.000	Severity=major	→	Sikll =Python,DB,Java
0.008	0.037	0.200	0.244	0.768	-0.002	Severity=critical	→	Sikll =DB,Java,Angular
0.006	0.122	0.051	0.951	2.498	0.004	Severity=blocker	→	Sikll =DB,Java,Angular
0.006	0.038	0.164	0.145	1.605	0.002	Severity=major	→	Sikll =C++,Angular,DB
0.005	0.200	0.025	16.250	0.492	-0.005	Severity=minor	→	Sikll =C,DB,HTML
0.005	0.025	0.200	0.081	1.537	0.002	Severity=critical	→	Sikll =java,C,DB
0.005	0.031	0.164	0.099	1.877	0.002	Severity=major	→	Sikll =java,C,DB
0.004	0.073	0.051	1.293	1.103	0.000	Severity=blocker	→	Sikll =Angular,DB,C
0.004	0.023	0.164	0.298	0.469	-0.004	Severity=major	→	Sikll =DB,Java,Angular

Figure 5: Assessment rules showing antecedent and consequent

The study finds out that one may require knowledge of C, databases, and HTML to resolve smaller problems, but major bugs may demand skills in DB, Java, and AngularJS. Exploring the opportunities of its pattern recognition, the algorithm provides the impression of what the severity of bugs can be compared to the appropriate skills, thereby aiding in a more efficient utilization of resources and more efficient handling of problems which is shown in FIGURE 5.

5. CONCLUSION

These outcomes point at the significance of aligning capabilities and bug allocations to enhance the problem-solving skills of team members and the exploitation of the resources at their disposal. When the skills of a team member perfectly fit the needs of a problem, his or her contribution to the success of the entire project and the performance of a single individual can be significant. The methods and inferences of this study can be applied to the bug management scenarios in various industries and with different sizes of the organization instead of being confined to the small business under study. The proposed approach offers a structured and information-oriented approach to enable more accurate and efficient decision-making when allocating tasks as software systems are increasingly becoming complex. Therefore, the article is a step toward more specific and productive bug assignment, which will eventually lead to improved software and customer satisfaction. Moreover, the researchers can advance this framework to research its implementation in the diverse areas where it is essential to assign tasks in an optimal manner. Such developments may further reinforce the role of intelligent task allocation in promoting productivity, consistency and quality of the outcome in a range of operational setups. To make this framework better in the future, additional components can be included such as learning curves, developer workload as well as changing skill sets. In dynamic projects, the use of advanced machine learning models can improve flexibility and accuracy in prediction.

Conflicts of interest

The authors of the manuscript mentioned that there is no potential conflict of interest.

References

[1] Jonsson L, Borg M, Broman D, Sandahl K, Eldh S, Runeson P. Automated Bug Assignment: Ensemble-Based Machine Learning in Large Scale Industrial Contexts. Empir Softw Eng. 2016;21:1533-1578.

[2] Paramesh SP, Ramya C, Shreedhara KS. Classifying the Unstructured IT Service Desk Tickets Using Ensemble of Classifiers. In 2018 3rd international conference on computational systems and information technology for sustainable solutions (CSITSS). IEEE. 2018:221-227.

[3] Goyal A, Sardana N. Analytical Study on Bug Triaging Practices. Inresearch Anthology on Recent Trends, Tools, and Implications of Computer Programming. IGI Global. 2021:1068-1094.

[4] Kulkarni RH, Padmanabham P. Integration of Artificial Intelligence Activities in Software Development Processes and Measuring Effectiveness of Integration. IET Softw .2017;11:18-

26.

[5] Yang C, Chen J, Fan X, Jiang J, Sun J. Silent Compiler Bug De-Duplication via Three-Dimensional Analysis. In Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis 2023:677-689.

[6] Tan Y, Xu S, Wang Z, Zhang T, Xu Z et al. Bug Severity Prediction Using Question-And-Answer Pairs From Stack Overflow. J Syst Softw. 2020;165:110567.

[7] Devaiya DT, Anvik J, Omee FY, Bheree M. CASTR: Assisting Bug Report Assignment Recommender Creation. International Conference on Software Engineering and Knowledge Engineering. 2021:639.

[8] Servant F, Jones JA. Whosefault: Automatic Developer-To-Fault Assignment Through Fault Localization. In 2012 34th International conference on software engineering (ICSE). IEEE. 2012:36-46.

[9] Bhattacharya P, Neamtiu I, Shelton CR. Automated, Highly-Accurate, Bug Assignment Using Machine Learning and Tossing Graphs. J Syst Softw. 2012;85:2275-2292.

[10] Xuan J, Jiang H, Ren Z, Yan J, Luo Z. Automatic Bug Triage Using Semi-Supervised Text Classification. 2017. ArXiv preprint: https://arxiv.org/pdf/1704.04769.

[11] Bhattacharya S, Bhatnagar V. Fuzzy Data Mining: A Literature Survey and Classification Framework. Int J Network. Virtual Organ. 2012;11:382-408.

[12] Sanchez DE, de Barros LC, Esmi E. On Interactive Fuzzy Boundary Value Problems. Fuzzy Sets Syst. 2019;358:84-96.

[13] Guizzo G, Vergilio SR, Pozo AT. Evaluating a Multi-Objective Hyper-Heuristic for the Integration and Test Order Problem. In 2015 Brazilian conference on intelligent Systems (BRACIS). 2015:1-6. IEEE.

[14] Bohlouli M, Mittas N, Kakarontzas G, Theodosiou T, Angelis L, et al. Competence Assessment as an Expert System for Human Resource Management: A Mathematical Approach. Expert Syst Appl. 2017;70:83-102.

[15] Yin Y, Dong X, Xu T. Rapid and Efficient Bug Assignment Using Elm for IoT Software. IEEE Access. 2018;6:52713-52724.

[16] Xi SQ, Yao Y, Xiao XS, Xu F, Lv J. Bug Triaging Based on Tossing Sequence Modelling. J Comput Sci Technol. 2019;34:942-956.

[17] Lavazza L, Morasca S. On the Evaluation of Effort Estimation Models. In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. 2017:41-50.