# **Appendix B: Systematic Literature Review**

**A. Protocol for Systematic Review:** The following protocol is formulated for this purpose.

**i. Terms:** Search string is used for searches in databases for a systematic literature review. Narrow and good search terms are useful for finding relevant studies (Brereton et al., 2007). Table B.1 shows the search string for the systematic review.

| **Search String** | |
| --- | --- |
| **Full Boolean Expression** for studies discussing BP tasks | (Software)  AND  ("Software bug" OR bug OR "Software Defect" OR Defect OR “Bug Characteristics”)  AND  (Prioritization OR Priority OR Prioritize OR Prioritized)  AND  (Triage OR Triaged OR Triaging OR Triages OR “Triage Meeting”) |

Table B.1 describes the search string.

**ii. Sources:** To identify sources of relevant studies for our systematic literature review, the following types of search methods are used: bibliographical databases, searching key journals and conference proceedings, search engines, backwards snowballing and grey literature.

**iii. Library and Search Engines:** Digital libraries and academic search engines are preferred sources for finding relevant literature therefore seven digital libraries and one search engine are used for our systematic literature review which is listed in Table B.2.

Table B.2 shows the sources for finding relevant literature.

| **S. No.** | **Digital Library** |
| --- | --- |
| 1 | IEEE Xplore |
| 2 | Springer Link |
| 3 | ACM |
| 4 | Scopus |
| 5 | Science Direct |
| 6 | ProQuest |
| 7 | EBSCo discovery Service |
| 8 | Google Scholar |

**iv. Backward Snowballing:** A backward snowballing technique is used for searching the new studies from the references list to identify other relevant literature. This is an alternative technique for a systematic approach that does not use search strings for database searches. This strategy is used because of the relevance of the relationship between included studies and their citations.

**v. Inclusion and Exclusion:** Inclusion criteria include all research papers of conference, thesis and journals/magazines reporting on BP that are published until 2024. Excluded papers are not freely available in the IIUM repository, subscribed digital libraries and search engines. The following are the criteria for the inclusion of research studies for BP tasks that are displayed in Table B.3. The exclusion criteria of research studies for BP tasks is shown in Table B.4.

Table B.3 shows the inclusion criteria.

| **Types** | All research papers of   1. Conference and journals. 2. Selected white papers. |
| --- | --- |
| All published empirical studies. |
| **Period** | All available studies until Feb 2024. |
| **Domain** | 1. Research papers published specifically on BP tasks. 2. Research papers published on other bug ug handling process that reports BP tasks.   Following are the exclusion criteria of research studies for BP tasks shown in Table B.4. |
| **Method** | All qualitative, and quantitative including theoretical, experiments and industrial case studies. |

Table B.4 describes exclusion criteria.

| **Domain** | Exclude software bugs triaging research papers on   * Developer prioritization and assignment of bug fixation. * Determine similar and duplicate bugs. * Determining bug severity. * And any other activity of Bug triaging meeting. |
| --- | --- |
| Exclude research papers on   * Prediction of software bug fixation time. * Detecting software bugs automatically. * Finding software bug locations. * Failure prediction. * Bug report categorization. |
| **Language** | Exclude non-English papers. |
| **Paper Type** | Excludes books, unpublished, or incomplete studies. |

**vi. Study Selection Procedures:** The following are the study selection procedures for Preliminary, Primary and Secondary Studies. Table B.5. describes the four-study selection procedure that will be performed for systematic review.

Table B.5. describes four study selection procedures of paper extraction using a systematic literature review protocol.

| **Preliminary Studies** | This includes random searches from different digital libraries and Google Scholar. Preliminary study was conducted to understand the BP domain and establish an initial problem statement to understand the phenomena of BP tasks. |
| --- | --- |
| **Primary Studies** | This includes seven digital libraries and one search engine for retrieving and selecting/filtering studies which includes IEEE Xplore, Springer Link, ACM, Scopus, Science Direct, ProQuest, EBSCo discovery Service and Google Scholar. In the first step, **1462** studies are retrieved. |
| **Secondary Studies** | Retrieved studies are filtered in a three-step process for further inclusion and exclusion. In this step, the papers are filtered on the basis of relevancy from their title and abstract. If the title and abstract were irrelevant, the paper was excluded at this stage. For many papers, abstracts and conclusions are reviewed to filter them where the title is showing relevancy. Finally, **169** papers are filtered based on the relevancy of their abstract or conclusion. In the second stage, the abstract and conclusion of 1**69** papers are reviewed. In the third step backward snowballing technique is employed. A total of 78 studies are selected from them based on their relevance with BP. |
| **Backward Snow Balling Technique** | From the pool of selected papers, studies specifically on BP tasks are retrieved using the backward snowballing technique. |

**B. Extraction of Data:**

**i. Number of Papers Retrieved, Filtered, and Selection for Review:** The following number of papers are retrieved and filtered, and relevant papers are included. Table B.6. and Figure B.1 shows these tabular and year-wise statistics.

Table B.6. shows statistics of papers’ extraction using SLR protocol.

| Main String | IEEE | | Springer | | ACM | | Scopus | | ProQuest | | EBSCo Discovery Service | | Science Direct | | Google Scholar | Total  (R, F and D) | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | R\* | F\* | R\* | F\* | R\* | F\* | R\* | F\* | R\* | F\* | R\* | F\* | R\* | F\* | F\*\* | R\* | F\* | S & I |
| Total Paper Till 2018 | 461 | 92 | 83 | 09 | 117 | 17 | 222 | 33 | 302 | 10 | 195 | 06 | 143 | 3 | 16 | 1462 | 169 | 78 |
| D\* | 0 | | 0 | | 15 | | 58 | | 0 | | 37 | | 8 | |  |  | |  |
| \* R= Available and retrieved abstracts from digital libraries against our search query  S=Selected relevant paper for review  \*\* F=Filtered for review in the first step/stage based on relevancy in the abstract  I=Included in Proposal D=Duplicate | | | | | | | | | | | | | | | | | | |

**A graph with numbers and a number of pages

Description automatically generated with medium confidence**



Diagram

Description automatically generated

Figure B.2 illustrates the process for relevant paper extractions.

ii. **Question for Data Extraction Form:** The questions for data extraction are given below. Table B.7 displays the data extraction form in which the data is extracted from the selected papers while Figure B.2 illustrates the process for extraction of relevant papers. .

Table B.7 displays the data extraction form.

| **Data Item** | **Value** |
| --- | --- |
| **Meta Data of Study** |  |
| Paper ID | Sample: P1, P2, P3 |
| Title |  |
| Author |  |
| Type of Study | Conference/Journal Name |
| Year of Publication |  |
| **Types of Studies (Q1)** | |
| What is the total number of studies on BP? |  |
| What are the major aspects of categorizing the studies on BP? |  |
| **Phenomena of BP (Q2)** |  |
| How is decision-problem modelled? |  |
| a. Is the decision problem modelled explicitly in the study? |  |
| b. Is the decision problem addressed effectively in the study? |  |
| **Research Challenges (Q3)** | |
| Challenges for BP |  |
| **DM Framework for Addressing BP (Q4 and Q5)** | |
| - Has the study formulated any Theory for DM to address the decision problem of BP? - Is any descriptive or prescriptive DM model created for BP? - Is the Proposed DM model rational or bounded rational?  Explain it in case for the study in case the answer is Yes. |  |
| Is study proposed the any automated technique, mechanism, tool, model, framework, or guideline of DM for addressing the decision problem of BP? Explain it in case for the study in case the answer is Yes. |  |
| Is the Proposed Technique, Tool, Model, Framework, or Guideline in Practice? Explain it in case for the study in case the answer is Yes. |  |
| **Factor Influence DMP for BP (Q6)** | |
| What factors have a relationship with BP? |  |

**C. Bibliography Reference**

Table B.8 illustrates bibliographic references, and assigns ID to each study in ascending order

| **ID** | **Bibliographic Reference** | |
| --- | --- | --- |
| P1 | Zheng, J., Williams, L., Nagappan, N., Snipes, W., Hudepohl, J. P., & Vouk, M. A. (2006). On the value of static analysis for fault detection in software. IEEE transactions on software engineering, 32(4), 240-253. | |
| P2 | Dalle, J. M., & Besten, M. D. (2010, May). Voting for bugs in Firefox: a voice for Mom and Dad?. In IFIP International Conference on Open Source Systems (pp. 73-84). Springer, Berlin, Heidelberg. | |
| P3 | [Guo, P. J., Zimmermann, T., Nagappan, N., & Murphy, B. (2010). Characterizing and predicting which bugs get fixed. Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10. https://doi.org/10.1145/1806799.1806871](about:blank) | |
| P4 | Yu, L., Tsai, W. T., Zhao, W., & Wu, F. (2010). Predicting defect priority based on neural networks. In International Conference on Advanced Data Mining and Applications (pp. 356-367). Springer, Berlin, Heidelberg. | |
| P5 | Soner, S., Jain, A., Tripathi, A., & Litoriya, R. (2010). A novel approach to calculate the severity and priority of bugs in software projects. In 2010 2nd International conference on education technology and computer (Vol. 2, pp. V2-50). IEEE. | |
| P6 | Kanwal, J., & Maqbool, O. (2010). Managing open bug repositories through bug report prioritization using SVMs. In Proceedings of the International Conference on Open-Source Systems and Technologies, Lahore, Pakistan (pp. 22-24). | |
| P7 | Khomh, F., Chan, B., Zou, Y., & Hassan, A. E. (2011). An entropy evaluation approach for triaging field crashes: A case study of mozilla firefox. In *2011 18th Working Conference on Reverse Engineering* (pp. 261-270). IEEE. | |
| P8 | [Sharma, M., Bedi, P., Chaturvedi, K., & Singh, V. (2012). Predicting the priority of a reported bug using machine learning techniques and cross project validation. 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA) IEEE. https://doi.org/10.1109/isda.2012.6416595](about:blank) | |
| P9 | Thung, F., Lo, D., Jiang, L., Rahman, F., & Devanbu, P. T. (2012, September). When would this bug get reported?. In 2012 28th IEEE International Conference on Software Maintenance (ICSM) (pp. 420-429). IEEE. | |
| P10 | Cohen J, Ferguson R, Hayes W. (2013). A Defect Prioritization Method Based on the Risk Priority Number. White Paper. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST. | |
| P11 | [Kaushik, N., Amoui, M., Tahvildari, L., Liu, W., & Li, S. (2013). Defect Prioritization in the Software Industry: Challenges and Opportunities. 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation. https://doi.org/10.1109/icst.2013.40](about:blank) | |
| P12 |  | |
| P13 | [Tian, Y., Lo, D., & Sun, C. (2013). DRONE: Predicting priority of reported bugs by multi-factor analysis. 2013 IEEE International Conference on Software Maintenance. https://doi.org/10.1109/icsm.2013.31](https://doi.org/10.1109/icsm.2013.31) | |
| P14 | [Alenezi, M., & Banitaan, S. (2013). Bug reports prioritization: Which features and classifier to use? 2013 12th International Conference on Machine Learning and Applications. https://doi.org/10.1109/icmla.2013.114](https://doi.org/10.1109/icmla.2013.114) | |
| P15 | [Sharma, M., Bedi, P., & Singh, V. B. (2014). An empirical evaluation of cross project priority prediction. International Journal of System Assurance Engineering and Management, 5(4), 651-663. https://doi.org/10.1007/s13198-014-0219-4](https://doi.org/10.1007/s13198-014-0219-4) | |
| P16 | Sharma, M., Bedi, P., & Singh, V. B. (2014). An empirical evaluation of cross project priority prediction. International Journal of System Assurance Engineering and Management, 5(4), 651-663. | |
| P17 | An, L., & Khomh, F. (2015). An empirical study of highly Impactful bugs in Mozilla projects. 2015 IEEE International Conference on Software Quality, Reliability and Security. https://doi.org/10.1109/qrs.2015.45 | |
| P18 | [Tian, Y., Lo, D., Xia, X., & Sun, C. (2015). Automated prediction of bug report priority using multi-factor analysis. Empirical Software Engineering, 20(5), 1354-1383. https://doi.org/10.1007/s10664-014-9331-y](about:blank) | |
| P19 | Zhang, J., Wang, X., Hao, D., Xie, B., Zhang, L., & Mei, H. (2015). A survey on bug-report analysis. Science China Information Sciences, 58(2), 1-24. <https://doi.org/10.1007/s11432-014-5241-2> | |
| P20 | [Ohira, M., Kashiwa, Y., Yamatani, Y., Yoshiyuki, H., Maeda, Y., Limsettho, N., Fujino, K., Hata, H., Ihara, A., & Matsumoto, K. (2015). A dataset of high impact bugs: Manually-classified issue reports. 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories. https://doi.org/10.1109/msr.2015.78](about:blank) | |
| P21 | Saha, R. K., Khurshid, S., & Perry, D. E. (2015). Understanding the triaging and fixing processes of long lived bugs. Information and software technology, 65, 114-128. | |
| P22 | Cabot, J., Izquierdo, J. L. C., Cosentino, V., & Rolandi, B. (2015, March). Exploring the use of labels to categorize issues in open-source software projects. In 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER) (pp. 550-554). IEEE. | |
| P23 | Dreyton, D., Araújo, A. A., Dantas, A., Freitas, Á., & Souza, J. (2015, September). Search-based bug report prioritization for kate editor bugs repository. In International Symposium on Search Based Software Engineering (pp. 295-300). Springer, Cham. | |
| P24 | Goyal, N., Aggarwal, N., & Dutta, M. (2015). A novel way of assigning software bug priority using supervised classification on clustered bugs data. In Advances in Intelligent Informatics (pp. 493-501). Springer, Cham. | |
| P25 | [Da Costa, D. A., McIntosh, S., Kulesza, U., & Hassan, A. E. (2016). The impact of switching to a rapid release cycle on the integration delay of addressed issues. Proceedings of the 13th International Workshop on Mining Software Repositories - MSR '16. https://doi.org/10.1145/2901739.2901764](about:blank) | |
| P26 | [Feng, Y., Jones, J. A. Feng, Chen, Z., & Fang, C. (2016). Multi-objective test report prioritization using image understanding. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering - ASE 2016. https://doi.org/10.1145/2970276.2970367](about:blank) | |
| P27 | [Savor, T., Douglas, M., Gentili, M., Williams, L., Beck, K., & Stumm, M. (2016). Continuous deployment at Facebook and OANDA. Proceedings of the 38th International Conference on Software Engineering Companion - ICSE '16. https://doi.org/10.1145/2889160.2889223](about:blank) | |
| P28 | [Uddin, J., Ghazali, R., Deris, M. M., Naseem, R., & Shah, H. (2016). A survey on bug prioritization. Artificial Intelligence Review, 47(2), 145-180. https://doi.org/10.1007/s10462-016-9478-6](about:blank) | |
| P29 | [Rocha, H., Valente, M. T., Marques-Neto, H., & Murphy, G. C. (2016). An Empirical Study on Recommendations of Similar Bugs. 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). https://doi.org/10.1109/saner.2016.87](about:blank) | |
| P30 | [Rabiger, S., Girisken, A., & Yilmaz, C. (2016). How to provide developers only with relevant information? 2016 7th International Workshop on Empirical Software Engineering in Practice (IWESEP). https://doi.org/10.1109/iwesep.2016.14](about:blank) | |
| P31 | Dreyton, D., Araújo, A. A., Dantas, A., Saraiva, R., & Souza, J. (2016, October). A multi-objective approach to prioritize and recommend bugs in open source repositories. In International Symposium on Search Based Software Engineering (pp. 143-158). Springer, Cham. | |
| P32 | Xu, Z., He, T., Zhang, W., Wang, Y., Liu, J., & Chen, Z. (2016, July). Exploring the Influence of Time Factor in Bug Report Prioritization. In SEKE (pp. 243-248). | |
| P33 | Coker, Z., Damevski, K., Le Goues, C., Kraft, N. A., Shepherd, D., & Pollock, L. (2017, September). Behavior Metrics for Prioritizing Investigations of Exceptions. In 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 554-563). IEEE. | |
| P34 | [Ko, A. J. (2017). A three-year participant observation of software startup software evolution. 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP). https://doi.org/10.1109/icse-seip.2017.29](about:blank) | |
| P35 | [Xie, M., Wang, Q., Yang, G., & Li, M. (2017). COCOON: Crowdsourced Testing Quality Maximization Under Context Coverage Constraint. 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE). https://doi.org/10.1109/issre.2017.25](about:blank) | |
| P36 | [Wan, Z., Lo, D., Xia, X., & Cai, L. (2017). Bug characteristics in blockchain systems: A large-scale empirical study. IEEE/ACM 14th International Conference on Mining Software Repositories (MSR). https://doi.org/10.1109/msr.2017.59](https://doi.org/10.1109/msr.2017.59) | |
| P37 | [Yusop, N. S., Grundy, J., & Vasa, R. (2017). Reporting usability defects: A systematic literature review. IEEE Transactions on Software Engineering, 43(9), 848-867. https://doi.org/10.1109/tse.2016.2638427](https://doi.org/10.1109/tse.2016.2638427) | |
| P38 | Bennin, K. E., Keung, J., Monden, A., Phannachitta, P., & Mensah, S. (2017, November). The significant effects of data sampling approaches on software defect prioritization and classification. In 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (pp. 364-373). IEEE. | |
| P39 | Choudhary, P. A., & Singh, S. (2017). Neural Network Based Bug Priority Prediction Model using Text Classification Techniques. International Journal of Advanced Research in Computer Science, 8(5). | |
| P40 | [Umer, Q., Liu, H., & Sultan, Y. (2018). Emotion based automated priority prediction for bug reports. IEEE Access, 6, 35743-35752. https://doi.org/10.1109/access.2018.2850910](about:blank) | |
| P41 | [Habayeb, M., Murtaza, S. S., Miranskyy, A., & Bener, A. B. (2018). On the use of hidden Markov model to predict the time to fix bugs. IEEE Transactions on Software Engineering, 44(12), 1224-1244. https://doi.org/10.1109/tse.2017.2757480](about:blank) | |
| P42 | Hillah, N. (2018). A Conceptual Tool to Improve the Management of Software Defects. In International Symposium on Business Modeling and Software Design (pp. 443-451). Springer, Cham. | |
| P43 | Akbarinasaji, S. (2018). Prioritizing lingering bugs. ACM SIGSOFT Software Engineering Notes, 43(1), 1-6. | |
| P44 | Wang, H., Zhou, M., Cheng, X., Chen, G., & Gu, M. (2018, November). Which defect should be fixed first? semantic prioritization of static analysis report. In International Conference on Software Analysis, Testing, and Evolution (pp. 3-19). Springer, Cham. | |
| P45 | Gao, G., Li, H., Chen, R., Ge, X., & Guo, S. (2018, October). Identification of high priority bug reports via integration method. In CCF Conference on Big Data (pp. 336-349). Springer, Singapore. | |
| P46 | [Kumari, M., & Singh, V. B. (2019). An Improved Classifier Based on Entropy and Deep Learning for Bug Priority Prediction. Advances in Intelligent Systems and Computing, 571-580. Springer, Cham. https://doi.org/10.1007/978-3-030-16657-1\_53](about:blank) | |
| P47 | [Zhang, W., & Challis, C. (2019). Automatic bug priority prediction using DNN based regression. Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery, 333-340. Springer, Cham. https://doi.org/10.1007/978-3-030-32456-8\_36](https://doi.org/10.1007/978-3-030-32456-8_36) | |
| P48 | Gomes, L. A. F., da Silva Torres, R., & Côrtes, M. L. (2019). Bug report severity level prediction in open source software: A survey and research opportunities. Information and software technology, 115, 58-78. | |
| P49 | Abou Khalil, Z., Constantinou, E., Mens, T., Duchien, L. and Quinton, C., 2019, September. A longitudinal analysis of bug handling across eclipse releases. In 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 1-12). IEEE. | |
| P50 | Noei, E., Zhang, F., Wang, S., & Zou, Y. (2019). Towards prioritizing user-related issue reports of mobile applications. Empirical Software Engineering, 24(4), 1964-1996. | |
| P51 | Gupta, V., Kumar, D., & Kapur, P. K. (2019). Optimizing the Defect Prioritization in Enterprise Application Integration. In Software Engineering (pp. 585-597). Springer, Singapore. | |
| P52 | Umer, Q., Liu, H., & Illahi, I. (2019). CNN-based automatic prioritization of bug reports. IEEE Transactions on Reliability, 69(4), 1341-1354. | |
| P53 | Yusop, N. S. M., Grundy, J., Schneider, J. G., & Vasa, R. (2020). How usability defects defer from non-usability defects?: A case study on open source projects. International Journal on Advanced Science, Engineering and Information Technology, 10(1), 98-105. | |
| P54 | Kumari, M., Singh, U. K., & Sharma, M. (2020, July). Entropy based machine learning models for software bug severity assessment in cross project context. In International Conference on Computational Science and Its Applications (pp. 939-953). Springer, Cham. | |
| P55 | Almhana, R., Ferreira, T., Kessentini, M., & Sharma, T. (2020). Understanding and Characterizing Changes in Bugs Priority: The Practitioners’ Perceptive. In 2020 IEEE 20th International Working Conference on Source Code Analysis and Manipulation (SCAM) (pp. 87-97). IEEE. | |
| P56 | Iqbal, S., Naseem, R., Jan, S., Alshmrany, S., Yasar, M., & Ali, A. (2020). Determining bug prioritization using feature reduction and clustering with classification. IEEE Access, 8, 215661-215678. | |
| P57 | Lenz, L., Felderer, M., Schwedes, S., & Müller, K. (2020). Explainable Priority Assessment of Software-Defects using Categorical Features at SAP HANA. In Proceedings of the Evaluation and Assessment in Software Engineering (pp. 366-367). | |
| P58 | Akbarinasaji, S., Kavaklioglu, C., Başar, A., & Neal, A. (2020). Partially observable Markov decision process to generate policies in software defect management. *Journal of Systems and Software*, *163*, 110518. | |
| P59 | Waqar, A. (2020). Software Bug Prioritization in Beta Testing Using Machine Learning Techniques. Journal of Computers for Society, 1(1), 24-34. | |
| P60 | Liu, Y., Ma, C., Dong, Z., Zhang, T., Cheng, J., & Zhang, J. (2020, April). Research on Defect Priority Classification of Crowdsourcing Testing for Mobile Applications. In Journal of Physics: Conference Series (Vol. 1518, No. 1, p. 012008). IOP Publishing. | |
| P61 | Izadi, M., Akbari, K., & Heydarnoori, A. (2020). Predicting the Objective and Priority of Issue Reports in a Cross project Context. arXiv e-prints, arXiv-2012. | |
| P62 | Yusop, N. S. M., Grundy, J., Schneider, J. G., & Vasa, R. (2020). How usability defects defer from non-usability defects?: A case study on open source projects. International Journal on Advanced Science, Engineering and Information Technology, 10(1), 98-105. | |
| P63 | Abou Khalil, Z., Constantinou, E., Mens, T., & Duchien, L. (2021). On the impact of release policies on bug handling activity: A case study of Eclipse. Journal of Systems and Software, 173, 110882. | |
| P64 | Almhana, R. (2021). Intelligent Software Bugs Localization, Triage and Prioritization (Doctoral dissertation). | |
| P65 | Almhana, R., & Kessentini, M. (2021). Considering dependencies between bug reports to improve bugs triage. Automated Software Engineering, 28(1), 1-26. | |
| P66 | Sharma, M., Kumari, M., & Singh, V. B. (2021). Bug priority assessment in cross-project context using entropy-based measure. In Advances in Machine Learning and Computational Intelligence (pp. 113-128). Springer, Singapore. | |
| P67 | Panichella, S., Canfora, G., & Di Sorbo, A. (2021). “Won’t We Fix this Issue?” Qualitative characterization and automated identification of won fix issues on GitHub. Information and Software Technology, 139, 106665. | |
| P68 | Malhotra, R., Dabas, A., Hariharasudhan, A. S., & Pant, M. (2021, January). A study on machine learning applied to software bug priority prediction. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 965-970). IEEE. | |
| P69 | Ahmed, H. A., Bawany, N. Z., & Shamsi, J. A. (2021). CapBug-a framework for automatic bug categorization and prioritization using NLP and machine learning algorithms. IEEE Access, 9, 50496-50512. | |
| P70 | Bani-Salameh, H., & Sallam, M. (2021). A deep-learning-based bug priority prediction using RNN-LSTM neural networks. e-Informatica Software Engineering Journal, 15(1). | |
| P71 | Goyal, A., & Sardana, N. (2021). Bug handling in service sector software. In Research Anthology on Recent Trends, Tools, and Implications of Computer Programming (pp. 1941-1960). IGI Global. | |
| P72 | Li, Z., Cai, G., Yu, Q., Liang, P., Mo, R., & Liu, H. (2024). Bug priority change: An empirical study on Apache projects. *Journal of Systems and Software*, *212*, 112019. | |
| P72 | | Gökçeoğlu, M., & Sözer, H. (2021). Automated defect prioritization based on defects resolved at various project periods. *Journal of Systems and Software*, *179*, 110993. |
| P73 | | Shatnawi, M. Q., & Alazzam, B. (2022). An Assessment of Eclipse Bugs' Priority and Severity Prediction Using Machine Learning. International Journal of Communication Networks and Information Security, 14(1), 62-69. |
| P74 | | Jahanshahi, H., Cevik, M., Navas-Sú, J., Başar, A., & González-Torres, A. (2022). Wayback Machine: A tool to capture the evolutionary behavior of the bug reports and their triage process in open-source software systems. *Journal of Systems and Software*, *189*, 111308. |
| P75 | | Wang WY, Wu CH, He J. Clebpi: Contrastive learning for bug priority inference. Information and Software Technology. 2023 Dec 1;164:107302. |
| P76 | | Raja, S. A., Aziz, M. S. A., & Shah, A. (2023). Modeling the Workflow of Bug Prioritization Tasks Descriptively Using the Past Events. *International Journal on Perceptive and Cognitive Computing*, *9*(2), 14-24. |
| P77 | | Panda, R. R., & Nagwani, N. K. (2024). Software bug priority prediction technique based on intuitionistic fuzzy representation and class imbalance learning. *Knowledge and Information Systems*, *66*(3), 2135-2164. |
| P78 | | Li, Z., Cai, G., Yu, Q., Liang, P., Mo, R., & Liu, H. (2024). Bug priority change: An empirical study on Apache projects. Journal of Systems and Software, 212, 112019. |