Supplementary material for:

Chapter 3 "Inconsistent labels identification based on cross-version source code comparison"

Appendix E. To what extent might previous studies be potentially influenced by inconsistent labels?

Our experimental results show that existing multi-version-project defect data sets (i.e. ECLIPSE-2007 [A1], Metrics-Repo-2010 [A2], JIRA-HA-2019 [A5], JIRA-RA-2019 [A5], and IND-JLMIV+R-2020 [A6]) contain inconsistent labels. In particular, for a defect prediction model, the existence of such inconsistent labels may considerably change its prediction ability, evaluation, and model interpretation. This raises concerns on the reliability of the experimental results or conclusions reported in previous studies that used these data sets. Therefore, the following question naturally arises: how many previous studies might be potentially influenced by these multi-version-project defect data sets? In this section, we answer this question by investigating the number of previous studies that used them as the subject data sets to conduct their experiments.

In order to avoid ambiguity, we separate the two concepts of "number of citations" and "number of literatures that had experimented with the collected data sets". The "number of citations" refers to the total number of citations by other literatures for the original papers that published the target multi-version-project defect data set. The "number of literatures that had experimented with the collected data sets" refers to the total number of other literatures, which not only cited the original papers of the target multi-version-project defect data set, but also used the target multi-version-project defect data set in their experiments. Generally, "number of citations" cannot be used as a proxy for the frequency of a data set usage, because other literatures may only introduce the methods or ideas of the cited paper. Therefore, we additionally count the "number of literatures that had experimented with the collected data sets" as a proxy for the frequency of a data set usage.

Table 8 summarizes the "number of citations" and "number of literatures that had experimented with the collected data sets" for the existing multi-version-project defect data sets investigated in our study. The first column lists the data sets. The second column lists the original paper(s) publishing each multi-version-project defect data set. The third column reports how many other literatures cite the original literature, i.e., "number of citations" (reported by Google scholar, March 26, 2021). The fourth column reports the total number of other literatures (written in English) that use the corresponding data sets to conduct their experiments, i.e., "number of literatures that had experimented with the collected data sets" (inspected by the first author and confirmed by the seventh author). Note that the Metrics-Repo-2010 data set was first published in [A2]. However, most literature cite [A3] and [A4] as its source. The reason was that [A3] was published in a well-known international conference on Predictive Models in Software Engineering (PROMISE), aiming to share publicly accessible data sets. In particular, Metrics-Repo-2010 was put on the corresponding promise repository website [A4]. Given this situation, we use them (i.e. [A2], [A3], and [A4]) as three sources to count the number of (different) citations. As can be seen, JIRA-RA-2019 (JIRA-HA-2019) and IND-JLMIV+R-2020 data sets were used by few studies (the "number of literatures that had experimented with the collected data sets" are 6 and 5 respectively), as they are two recently published multi-version-project defect data sets. However, ECLIPSE-2007 and Metrics-Repo-2010 data sets were widely used in previous studies (the "number of literatures that had experimented with the collected data sets" is 144 and 264 respectively). This indicates that inconsistent labels have a potentially wide influence on previous studies. At the bottom of Table 8, we list all the literatures we investigated.

It is important to note that the influence of inconsistent labels on the existing literature may be overestimated, although the "number of literatures that had experimented with the collected data sets" is a more secure proxy for estimating the potential influence of inconsistent labels than the "number of citations". This is because our study does not conduct replication experiments to investigate the specific influence of inconsistent labels on each of the existing literatures (it is a large amount of work that could not be done in our study alone). It is still an open problem to investigate the actual influence of each multi-version-project defect data set with inconsistent labels on the existing literatures, pending an in-depth or extensive empirical study in the future.

TABLE 8

Literatures potentially influenced by target multi-version-project defect data sets

Defect data sets	Source	Number of citations	Number of literatures that had experimented with the target data sets	List number range of literatures that had experimented with the target data sets
ECLIPSE-2007	[A1]	817	144	[1~144]
Metrics-Repo-2010	[A2, A3, A4]	453	264	[145~408]
JIRA-HA-2019 / JIRA-RA-2019	[A5]	17	6	[409~414]
IND-JLMIV+R-2020	[A6]	9	5	[415~419]

The literatures of five target multi-version-project defect data sets

[A1] T. Zimmermann, R. Premraj, A. Zeller. Predicting defects for Eclipse. In Proceedings of the Third International Workshop on Predictor Models in Software Engineering, ser. PROMISE '07. IEEE Computer Society, 2007: 9–.

[A2] M. Jureczko, D. Spinellis. Using object-oriented design metrics to predict software defects. In Models and Methods of System Dependability. Oficyna Wydawnicza Politechniki Wrocławskiej, 2010: 69-81.

[A3] M. Jureczko, L. Madeyski. Towards identifying software project clusters with regard to defect prediction. In: Proceedings of the 6th International Conference on Predictive Models in Software Engineering, 2010: 1–10.

[A4] T. Menzies, B. Caglayan, E. Kocaguneli, J. Krall, F. Peters, B. Turhan. The promise repository of empirical software engineering data, June 2012.

[A5] S. Yatish, J. Jiarpakdee, P. Thongtanunam, C. Tantithamthavorn. Mining software defects: should we consider affected releases? ICSE 2019: 654-665.

[A6] S. Herbold, A. Trautsch, F. Trautsch. Issues with SZZ: An empirical assessment of the state of practice of defect prediction data collection. arXiv preprint arXiv:1911.08938v2, 2020.

The list of citations using the ECLIPSE-2007 data set

[1] Moser R, Pedrycz W, Succi G. A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction[C]//Proceedings of the 30th international conference on Software engineering. ACM, 2008: 181-190.

[2] Zimmermann T, Nagappan N, Gall H, et al. Cross-project defect prediction: a large scale experiment on data vs. domain vs. process[C]//Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. ACM, 2009: 91-100.

[3] Bird C, Bachmann A, Aune E, et al. Fair and balanced?: bias in bug-fix datasets[C]//Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. ACM, 2009: 121-130.

[4] Nagappan N, Zeller A, Zimmermann T, et al. Change bursts as defect predictors[C]//2010 IEEE 21st International Symposium on Software Reliability Engineering. IEEE, 2010: 309-318.

[5] Mende T, Koschke R. Effort-aware defect prediction models[C]//2010 14th European Conference on Software Maintenance and Reengineering. IEEE, 2010: 107-116.

[6] Tantithamthavorn C, McIntosh S, Hassan A E, et al. Automated parameter optimization of classification techniques for defect prediction models[C]//2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE). IEEE, 2016: 321-332.

[7] Li M, Zhang H, Wu R, et al. Sample-based software defect prediction with active and semi-supervised learning[J]. Automated Software Engineering, 2012, 19(2): 201-230.

[8] Zhou Y, Xu B, Leung H. On the ability of complexity metrics to predict fault-prone classes in object-oriented systems[J]. Journal of Systems and Software, 2010, 83(4): 660-674.

[9] Tantithamthavorn C, McIntosh S, Hassan A E, et al. An empirical comparison of model validation techniques for defect prediction models[J]. IEEE Transactions on Software Engineering, 2017, 43(1): 1-18.

[10] Zhang H. An investigation of the relationships between lines of code and defects[C]//2009 IEEE International Conference on Software Maintenance. IEEE, 2009: 274-283.

[11] Khoshgoftaar T M, Gao K, Seliya N. Attribute selection and imbalanced data: Problems in software defect prediction[C]//2010 22nd IEEE International Conference on Tools with Artificial Intelligence. IEEE, 2010, 1: 137-144.

[12] Shihab E, Jiang Z M, Ibrahim W M, et al. Understanding the impact of code and process metrics on post-release defects: a case study on the eclipse project[C]//Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, 2010: 4.

[13] Zhang H. On the distribution of software faults[J]. IEEE Transactions on Software Engineering, 2008, 34(2): 301-302.

[14] Bettenburg N, Hassan A E. Studying the impact of social structures on software quality[C]//2010 IEEE 18th International Conference on Program Comprehension. IEEE, 2010: 124-133.

[15] Wang H, Khoshgoftaar T M, Napolitano A. A comparative study of ensemble feature selection techniques for software defect prediction[C]//2010 Ninth International Conference on Machine Learning and Applications. IEEE, 2010: 135-140.

[16] Yang X, Tang K, Yao X. A learning-to-rank approach to software defect prediction[J]. IEEE Transactions on Reliability, 2015, 64(1): 234-246.

[17] Nguyen T H D, Adams B, Hassan A E. Studying the impact of dependency network measures on software quality[C]//2010 IEEE International Conference on Software Maintenance. IEEE, 2010: 1-10.

[18] Premraj R, Herzig K. Network versus code metrics to predict defects: A replication study[C]//2011 International Symposium on Empirical Software Engineering and Measurement. IEEE, 2011: 215-224.

[19] Wang H, Khoshgoftaar T M, Seliya N. How many software metrics should be selected for defect prediction?[C]//Twenty-Fourth International FLAIRS Conference. 2011.

[20] Kpodjedo S, Ricca F, Galinier P, et al. Design evolution metrics for defect prediction in object oriented systems[J]. Empirical Software Engineering, 2011, 16(1): 141-175.

[21] Rodriguez D, Herraiz I, Harrison R. On software engineering repositories and their open problems[C]//2012 First International

Workshop on Realizing AI Synergies in Software Engineering (RAISE). IEEE, 2012: 52-56.

[22] Liu W, Liu S, Gu Q, et al. Empirical studies of a two-stage data preprocessing approach for software fault prediction[J]. IEEE Transactions on Reliability, 2016, 65(1): 38-53.

[23] Khoshgoftaar T M, Gao K, Napolitano A, et al. A comparative study of iterative and non-iterative feature selection techniques for software defect prediction[J]. Information Systems Frontiers, 2014, 16(5): 801-822.

[24] Wang H, Khoshgoftaar T M, Van Hulse J. A comparative study of threshold-based feature selection techniques[C]//2010 IEEE International Conference on Granular Computing. IEEE, 2010: 499-504.

[25] Khoshgoftaar T M, Gao K, Napolitano A. An empirical study of feature ranking techniques for software quality prediction[J]. International journal of software engineering and knowledge engineering, 2012, 22(02): 161-183.

[26] Elish M O, Al-Yafei A H, Al-Mulhem M. Empirical comparison of three metrics suites for fault prediction in packages of object-oriented systems: A case study of Eclipse[J]. Advances in Engineering Software, 2011, 42(10): 852-859.

[27] Zhou Y, Xu B, Leung H, et al. An in-depth study of the potentially confounding effect of class size in fault prediction[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2014, 23(1): 10.

[28] Caglayan B, Bener A, Koch S. Merits of using repository metrics in defect prediction for open source projects[C]//2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development. IEEE, 2009: 31-36.

[29] Wang H, Khoshgoftaar T M, Napolitano A. Software measurement data reduction using ensemble techniques[J]. Neurocomputing, 2012, 92: 124-132.

[30] Li W, Huang Z, Li Q. Three-way decisions based software defect prediction[J]. Knowledge-Based Systems, 2016, 91: 263-274.

[31] Shihab E. An exploration of challenges limiting pragmatic software defect prediction[D]. , 2012.

[32] Zeller A, Zimmermann T, Bird C. Failure is a four-letter word: a parody in empirical research[C]//Proceedings of the 7th International Conference on Predictive Models in Software Engineering. ACM, 2011: 5.

[33] Zimmerman T, Nagappan N, Herzig K, et al. An empirical study on the relation between dependency neighborhoods and failures[C]//2011 Fourth IEEE International Conference on Software Testing, Verification and Validation. IEEE, 2011: 347-356.

[34] Krishnan S, Strasburg C, Lutz R R, et al. Are change metrics good predictors for an evolving software product line?[C]//Proceedings of the 7th International Conference on Predictive Models in Software Engineering. ACM, 2011: 7.

[35] Ibrahim W M, Bettenburg N, Adams B, et al. On the relationship between comment update practices and software bugs[J]. Journal of Systems and Software, 2012, 85(10): 2293-2304.

[36] Moser R, Pedrycz W, Succi G. Analysis of the reliability of a subset of change metrics for defect prediction[C]//Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. ACM, 2008: 309-311.

[37] Lu H, Kocaguneli E, Cukic B. Defect prediction between software versions with active learning and dimensionality reduction[C]//2014 IEEE 25th International Symposium on Software Reliability Engineering. IEEE, 2014: 312-322.

[38] Oyetoyan T D, Cruzes D S, Conradi R. A study of cyclic dependencies on defect profile of software components[J]. Journal of Systems and Software, 2013, 86(12): 3162-3182.

[39] Tantithamthavorn C, McIntosh S, Hassan A E, et al. The impact of automated parameter optimization on defect prediction models[J]. IEEE Transactions on Software Engineering, 2018:1-1.

[40] Rathore S S, Kumar S. Towards an ensemble based system for predicting the number of software faults[J]. Expert Systems with Applications, 2017, 82: 357-382.

[41] Wahyudin D, Ramler R, Biffl S. A framework for defect prediction in specific software project contexts[C]//IFIP Central and East European Conference on Software Engineering Techniques. Springer, Berlin, Heidelberg, 2008: 261-274.

[42] Gao K, Khoshgoftaar T M. Software Defect Prediction for High-Dimensional and Class-Imbalanced Data[C]//SEKE. 2011: 89-94.

[43] Wang H, Khoshgoftaar T M, Van Hulse J, et al. Metric selection for software defect prediction[J]. International Journal of Software Engineering and Knowledge Engineering, 2011, 21(02): 237-257.

[44] Pipitone J, Easterbrook S. Assessing climate model software quality: a defect density analysis of three models[J]. Geoscientific Model Development, 2012, 5(4): 1009-1022.

[45] Liu Y, Cheah W P, Kim B K, et al. Predict software failure-prone by learning Bayesian network[J]. International Journal of Advanced Science and Technology, 2008, 1(1): 35-42.

[46] Krishnan S, Strasburg C, Lutz R R, et al. Predicting failure-proneness in an evolving software product line[J]. Information and Software Technology, 2013, 55(8): 1479-1495.

[47] Tan X, Peng X, Pan S, et al. Assessing software quality by program clustering and defect prediction[C]//2011 18th working conference on Reverse Engineering. IEEE, 2011: 244-248.

[48] Chen J, Liu S, Liu W, et al. A two-stage data preprocessing approach for software fault prediction[C]//2014 Eighth International Conference on Software Security and Reliability (SERE). IEEE, 2014: 20-29.

[49] Marinescu C. Are the classes that use exceptions defect prone?[C]//Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution. ACM, 2011: 56-60.

[50] Gao K, Khoshgoftaar T M, Napolitano A. Impact of data sampling on stability of feature selection for software measurement data[C]//2011 IEEE 23rd International Conference on Tools with Artificial Intelligence. IEEE, 2011: 1004-1011.

[51] Rathore S S, Kumar S. Linear and non-linear heterogeneous ensemble methods to predict the number of faults in software systems[J]. Knowledge-Based Systems, 2017, 119: 232-256.

[52] Marinescu R, Marinescu C. Are the clients of flawed classes (also) defect prone?[C]//2011 IEEE 11th International Working Conference on Source Code Analysis and Manipulation. IEEE, 2011: 65-74.

[53] Tantithamthavorn C, Hassan A E. An experience report on defect modelling in practice: Pitfalls and challenges[C]//Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice. ACM, 2018: 286-295.

[54] Wang H, Khoshgoftaar T M, Napolitano A. An Empirical Study of Software Metrics Selection Using Support Vector Machine[C]//SEKE. 2011: 83-88.

[55] Gao K, Khoshgoftaar T M, Napolitano A. Combining Feature Subset Selection and Data Sampling for Coping with Highly Imbalanced Software Data[C]//SEKE. 2015: 439-444.

[56] Wang H, Khoshgoftaar T M, Wald R. Measuring robustness of feature selection techniques on software engineering datasets[C]//2011 IEEE International Conference on Information Reuse & Integration. IEEE, 2011: 309-314.

[57] Plosch R, Gruber H, Hentschel A, et al. On the relation between external software quality and static code analysis[C]//2008 32nd Annual IEEE Software Engineering Workshop. IEEE, 2008: 169-174.

[58] Marinescu C. Should we beware the exceptions? an empirical study on the eclipse project[C]//2013 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. IEEE, 2013: 250-257.

[59] Tantithamthavorn C, Hassan A E, Matsumoto K. The impact of class rebalancing techniques on the performance and interpretation of defect prediction models[J]. IEEE Transactions on Software Engineering, 2018.

[60] Liu W, Liu S, Gu Q, et al. Fecs: A cluster based feature selection method for software fault prediction with noises[C]//2015 IEEE 39th Annual Computer Software and Applications Conference. IEEE, 2015, 2: 276-281.

[61] Kuo C S, Huang C Y. A study of applying the bounded generalized pareto distribution to the analysis of software fault distribution[C]//2010 IEEE International Conference on Industrial Engineering and Engineering Management. IEEE, 2010: 611-615.

[62] Devine T, Goseva-Popstojanova K, Krishnan S, et al. Assessment and cross-product prediction of software product line quality: accounting for reuse across products, over multiple releases[J]. Automated Software Engineering, 2016, 23(2): 253-302.

[63] Zhang H, Cheung S C. A cost-effectiveness criterion for applying software defect prediction models[C]//Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. ACM, 2013: 643-646.

[64] Zhang H, Wu R. Sampling program quality[C]//2010 IEEE International Conference on Software Maintenance. IEEE, 2010: 1-10.

[65] Wang H, Khoshgoftaar T M, Liang Q. A study of software metric selection techniques: stability analysis and defect prediction model performance[J]. International journal on artificial intelligence tools, 2013, 22(05): 1360010.

[66] Pipitone J. Software quality in climate modelling[J]. Master's thesis, Department of Computer Science, University of Toronto, 2010.

[67] Steff M, Russo B. Measuring architectural change for defect estimation and localization[C]//2011 International Symposium on Empirical Software Engineering and Measurement. IEEE, 2011: 225-234.

[68] Choudhary G R, Kumar S, Kumar K, et al. Empirical analysis of change metrics for software fault prediction[J]. Computers & Electrical Engineering, 2018, 67: 15-24.

[69] Wang H, Khoshgoftaar T M, Napolitano A. An empirical investigation on wrapper-based feature selection for predicting software quality[J]. International Journal of Software Engineering and Knowledge Engineering, 2015, 25(01): 93-114.

[70] Wang H, Khoshgoftaar T M, Seliya N. On the stability of feature selection methods in software quality prediction: an empirical investigation[J]. International Journal of Software Engineering and Knowledge Engineering, 2015, 25(09n10): 1467-1490.

[71] Wang H, Khoshgoftaar T M, Napolitano A. Stability of filter-and wrapper-based software metric selection techniques[C]//Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014). IEEE, 2014: 309-314.

[72] Wang H, Khoshgoftaar T M, Napolitano A. An empirical study on the stability of feature selection for imbalanced software engineering data[C]//2012 11th International Conference on Machine Learning and Applications. IEEE, 2012, 1: 317-323.

[73] Khoshgoftaar T M, Gao K, Van Hulse J. Feature selection for highly imbalanced software measurement data[M]//Recent trends in information reuse and integration. Springer, Vienna, 2012: 167-189.

[74] GAO K, KHOSHGOFTAAR T M, WALD R. The use of under- and oversampling within ensemble feature selection and classification for software quality prediction[J]. International Journal of Reliability, Quality and Safety Engineering, 2014, 21(01): 1450004.

[75] Zhang H, Tan H B K, Marchesi M. The distribution of program sizes and its implications: An eclipse case study[C]//1st International Symposium on Emerging Trends in Software Metrics. 2009: 1-10.

[76] Khoshgoftaar T M, Gao K, Napolitano A. A Comparative Study of Different Strategies for Predicting Software Quality[C]//SEKE. 2011, 2011: 65-70.

[77] Mizuno O, Hata H. An empirical comparison of fault-prone module detection approaches: Complexity metrics and text feature metrics[C]//2010 IEEE 34th Annual Computer Software and Applications Conference. IEEE, 2010: 248-249.

[78] Wang H, Khoshgoftaar T M, Wald R, et al. A comparative study on the stability of software metric selection techniques[C]//2012 11th International Conference on Machine Learning and Applications. IEEE, 2012, 2: 301-307.

[79] Guo Y, Würsch M, Giger E, et al. An empirical validation of the benefits of adhering to the law of demeter[C]//2011 18th Working Conference on Reverse Engineering. IEEE, 2011: 239-243.

[80] Mizuno O, Hata H. An integrated approach to detect fault-prone modules using complexity and text feature metrics[M]//Advances in Computer Science and Information Technology. Springer, Berlin, Heidelberg, 2010: 457-468.

[81] Oyetoyan T D, Cruzes D S, Conradi R. Can refactoring cyclic dependent components reduce defect-proneness?[C]//2013 IEEE International Conference on Software Maintenance. IEEE, 2013: 420-423.

[82] Khoshgoftaar T M, Gao K, Napolitano A. An empirical study of predictive modeling techniques of software quality[C]//International Conference on Bio-Inspired Models of Network, Information, and Computing Systems. Springer, Berlin, Heidelberg, 2010: 288-302.

[83] Jiarpakdee J, Tantithamthavorn C, Treude C. AutoSpearman: Automatically Mitigating Correlated Software Metrics for Interpreting Defect Models[C]//2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2018: 92-103.

[84] Wang H, Khoshgoftaar T M. Measuring stability of threshold-based feature selection techniques[C]//2011 IEEE 23rd International Conference on Tools with Artificial Intelligence. IEEE, 2011: 986-993.

[85] Gao K, Khoshgoftaar T M, Napolitano A. The use of ensemble-based data preprocessing techniques for software defect prediction[J]. International Journal of Software Engineering and Knowledge Engineering, 2014, 24(09): 1229-1253.

[86] Khoshgoftaar T M, Gao K, Van Hulse J. A novel feature selection technique for highly imbalanced data[C]//2010 IEEE International Conference on Information Reuse & Integration. IEEE, 2010: 80-85.

[87] Khoshgoftaar T M, Gao K, Napolitano A. Improving software quality estimation by combining feature selection strategies with sampled ensemble learning[C]//Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014). IEEE, 2014: 428-433.

[88] Khoshgoftaar T M, Gao K, Napolitano A. Exploring an iterative feature selection technique for highly imbalanced data sets[C]//2012 IEEE 13th International Conference on Information Reuse & Integration (IRI). IEEE, 2012: 101-108.

[89] Mizuno O, Hata H. A metric to detect fault-prone software modules using text filtering[J]. International Journal of Reliability and Safety, 2013, 7(1): 17-31.

[90] Altidor W, Khoshgoftaar T M, Gao K. Wrapper-based feature

ranking techniques for determining relevance of software engineering metrics[J]. International Journal of Reliability, Quality and Safety Engineering, 2010, 17(05): 425-464.

[91] Kumari D, Rajnish K. Comparing Efficiency of Software Fault Prediction Models Developed Through Binary and Multinomial Logistic Regression Techniques[M]//Information Systems Design and Intelligent Applications. Springer, New Delhi, 2015: 187-197.

[92] Wang H, Khoshgoftaar T M, Wald R. Measuring stability of feature selection techniques on real-world software datasets[M]//Information Reuse and Integration in Academia and Industry. Springer, Vienna, 2013: 113-132.

[93] Oyetoyan T D, Cruzes D S, Conradi R. Transition and defect patterns of components in dependency cycles during software evolution[C]//2014 Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE). IEEE, 2014: 283-292.

[94] Jia H, Shu F, Yang Y, et al. Data transformation and attribute subset selection: Do they help make differences in software failure prediction?[C]//2009 IEEE International Conference on Software Maintenance. IEEE, 2009: 519-522.

[95] Wang H, Khoshgoftaaar T M, Liang Q. Stability and classification performance of feature selection techniques[C]//2011 10th International Conference on Machine Learning and Applications and Workshops. IEEE, 2011, 1: 151-156.

[96] Mahmood Z, Bowes D, Hall T, et al. Reproducibility and replicability of software defect prediction studies[J]. Information and Software Technology, 2018, 99: 148-163.

[97] Mizuno O. On effects of tokens in source code to accuracy of fault-prone module prediction[C]//2013 International Computer Science and Engineering Conference (ICSEC). IEEE, 2013: 103-108.

[98] Gao K, Khoshgoftaar T M, Napolitano A. An empirical investigation of combining filter-based feature subset selection and data sampling for software defect prediction[J]. International Journal of Reliability, Quality and Safety Engineering, 2015, 22(06): 1550027.

[99] Zimmermann T, Nagappan N, Williams L, et al. An empirical study of the factors relating field failures and dependencies[C]//IEEE Fourth International Conference on Software Testing, Verification and Validation. 2011: 347-356.

[100] Gao K, Khoshgoftaar T M, Napolitano A. Aggregating data sampling with feature subset selection to address skewed software defect data[J]. International Journal of Software Engineering and Knowledge Engineering, 2015, 25(09n10): 1531-1550.

[101] Mizuno O, Hirata Y. Fault-prone module prediction using contents of comment lines[C]//Proc. of International Workshop on Empirical Software Engineering in Practice 2010 (IWESEP2010). 2010, 12: 39-44.

[102] Krishnan S. Evidence-based defect assessment and prediction for software product lines[J]. 2013.

[103] Gao K, Khoshgoftaar T M. Assessments of Feature Selection Techniques with Respect to Data Sampling for Highly Imbalanced Software Measurement Data[J]. International Journal of Reliability, Quality and Safety Engineering, 2015, 22(02): 1550010.

[104] Muthukumaran K, Dasgupta A, Abhidnya S, et al. On the effectiveness of cost sensitive neural networks for software defect prediction[C]//International Conference on Soft Computing and Pattern Recognition. Springer, Cham, 2016: 557-570.

[105] Han W, Lung C H, Ajila S A. Empirical investigation of code and process metrics for defect prediction[C]//2016 IEEE Second International Conference on Multimedia Big Data (BigMM). IEEE, 2016: 436-439.

[106] Gao K, Khoshgoftaar T M, Napolitano A. Investigating two approaches for adding feature ranking to sampled ensemble learning for software quality estimation[J]. International Journal of Software Engineering and Knowledge Engineering, 2015, 25(01): 115-146.

[107] Kumari D, Rajnish K. Investigating the Effect of Object-oriented Metrics on Fault Proneness Using Empirical Analysis[J]. International Journal of Software Engineering and Its Applications, 2015, 9(2): 171-188.

[108] Maheshwari S, Agarwal S. Three-way decision based Defect Prediction for Object Oriented Software[C]//Proceedings of the International Conference on Advances in Information Communication Technology & Computing. ACM, 2016: 4.

[109] Kumari D, Rajnish K. Evaluating The Impact Of Binary And Multinomial LR In Developing Software Fault Prediction Model Using OO-Metrics[J]. Global Journal of Pure and Applied Mathematics, 2015, 11(1): 437-462.

[110] Bettenburg N. Studying the Impact of Developer Communication on the Quality and Evolution of a Software System[D]. , 2014.

[111] Shriram C K, Muthukumaran K, Bhanu Murthy N L. Empirical Study on the Distribution of Bugs in Software Systems[J]. International Journal of Software Engineering and Knowledge Engineering, 2018, 28(01): 97-122.

[112] Muthukumaran K, Srinivas S, Malapati A, et al. Software Defect Prediction Using Augmented Bayesian Networks[C]//International Conference on Soft Computing and Pattern Recognition. Springer, Cham, 2016: 279-293.

[113] Li H Y, Li M, Zhou Z H. Towards one reusable model for various software defect mining tasks[J].

[114] Jiarpakdee J, Tantithamthavorn C, Treude C. Artefact: An R Implementation of the AutoSpearman Function[C]//2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2018: 711-711.

[115] Wang H, Khoshgoftaar T M, Napolitano A. Choosing the Best Classification Performance Metric for Wrapper-based Software Metric Selection for Defect Prediction[C]//SEKE. 2014: 540-545.

[116] Buchari M A, Mardiyanto S, Hendradjaya B. Implementation of Chaotic Gaussian Particle Swarm Optimization for Optimize Learning-to-Rank Software Defect Prediction Model Construction[C]//Journal of Physics: Conference Series. IOP Publishing, 2018, 978(1): 012079.

[117] Nagappan M, Murphy B, Vouk M. Which code construct metrics are symptoms of post release failures?[C]//Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics. ACM, 2011: 65-68.

[118] Herraiz Tabernero I, Shihab E, Nguyen T H D, et al. Impact of Installation Counts on Perceived Quality: A Case Study on Debian[J]. 2011.

[119] Verma R, Gupta A. An approach of attribute selection for reducing false alarms[C]//2012 CSI Sixth International Conference on Software Engineering (CONSEG). IEEE, 2012: 1-7.

[120] Rathore S S, Kumar S. An Approach for the Prediction of Number of Software Faults Based on the Dynamic Selection of Learning Techniques[J]. IEEE Transactions on Reliability, 2018 (99): 1-21.

[121] Zeller A, Zimmermann T, Bird C. Failure is a four-letter word[J]. 2011.

[122] Hashem A. A COMPREHENSIVE EMPIRICAL VALIDATION OF PACKAGE-LEVEL METRICS FOR OO SYSTEMS[D]. King Fahd University of Petroleum and Minerals, 2010.

[123] Khoshgoftaar T M, Gao K, Chen Y, et al. Comparing Feature Selection Techniques for Software Quality Estimation Using Data-Sampling-Based Boosting Algorithms[J]. International Journal of Reliability, Quality and Safety Engineering, 2015, 22(03): 1550013.

[124]Wang H, Khoshgoftaar T M, Wald R, et al. A novel dataset-similarity-aware approach for evaluating stability of software metric selection techniques[C]//2012 IEEE 13th International Conference on Information Reuse & Integration (IRI). IEEE, 2012: 1-8.
[125]Jiarpakdee J, Tantithamthavorn C, Treude C. The impact of automated feature selection techniques on the interpretation of defect models[J]. Empirical Software Engineering, 2020: 1-49.
[126]Goseva-Popstojanova K, Ahmad M, Alshehri Y. Software fault proneness prediction with group lasso regression: On factors that affect classification performance[C]//2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). IEEE, 2019, 2: 336-343.
[127]Xu Z, Ye S, Zhang T, et al. MVSE: Effort-Aware Heterogeneous Defect Prediction via Multiple-View Spectral Embedding[C]//2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2019: 10-17.
[128]Huo X, Li M. On cost-effective software defect prediction: Classification or ranking?[J]. Neurocomputing, 2019, 363: 339-350.
[129]Pham V, Lokan C, Kasmarik K. A Better Set of Object-Oriented Design Metrics for Within-Project Defect Prediction[M]//Proceedings of the Evaluation and Assessment in Software Engineering. 2020: 230-239.
[130]Rahman A. Software Defect Prediction Using Rich Contextualized Language Use Vectors[D]. , 2019.
[131]Zheng W, Mo S, Jin X, et al. Software Defect Prediction Model Based on Improved Deep Forest and AutoEncoder by Forest[C]//SEKE. 2019: 419-540.
[132]Yang X. Evaluating Software Metrics for Sorting Software Modules in Order of Defect Count[C]//ICSOFT. 2019: 94-105.
[133]Li H Y, Li M, Zhou Z H. Towards one reusable model for various software defect mining tasks[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham, 2019: 212-224.
[134]Nair P R. Optimizing bug prediction in software testing using Super Learner[D]. Dublin, National College of Ireland, 2019.
[135]Bassuday K, Ahmed M. Fault Prediction in Android Systems through AI[J]. 2019.
[136]Tosun Misirli A, Murphy B, Zimmermann T, et al. An explanatory analysis on eclipse beta-release bugs through in-process metrics[C]//Proceedings of the 8th international workshop on Software quality. ACM, 2011: 26-33.
[137]Arshad A, Riaz S, Jiao L, et al. The empirical study of semi-supervised deep fuzzy C-mean clustering for software fault prediction[J]. IEEE Access, 2018, 6: 47047-47061.
[138]Kidwell B R. MiSFIT: Mining Software Fault Information and Types[J]. 2015.
[139]Shaikh M, Lee K S, Lee C G. Assessing the Bug-Prediction with Re-Usability Based Package Organization for Object Oriented Software Systems[J]. IEICE TRANSACTIONS on Information and Systems, 2017, 100(1): 107-117.
[140]Ferenc R, Tóth Z, Ladányi G, et al. A public unified bug dataset for java and its assessment regarding metrics and bug prediction[J]. Software Quality Journal, 2020: 1-60.
[141]Ferenc R, Siket I, Hegedűs P, et al. Employing Partial Least Squares Regression with Discriminant Analysis for Bug Prediction[J]. arXiv preprint arXiv:2011.01214, 2020.
[142]Wu Y, Yao J, Chang S, et al. LIMCR: Less-Informative Majorities Cleaning Rule Based on Naïve Bayes for Imbalance Learning in Software Defect Prediction[J]. Applied Sciences, 2020, 10(23): 8324.
[143]Rajbahadur G. UNDERSTANDING THE IMPACT OF EXPERIMENTAL DESIGN CHOICES ON MACHINE LEARNING CLASSIFIERS IN SOFTWARE ANALYTICS[D].
[144]Pham V, Lokan C, Kasmarik K. A Better Set of Object-Oriented Design Metrics for Within-Project Defect Prediction[M]//Proceedings of the Evaluation and Assessment in Software Engineering. 2020: 230-239.

The list of citations using the Metrics-Repo-2010 data set

[145]He Z, Shu F, Yang Y, et al. An investigation on the feasibility of cross-project defect prediction[J]. Automated Software Engineering, 2012, 19(2): 167-199.
[146]Peters F, Menzies T, Marcus A. Better cross company defect prediction[C]//Proceedings of the 10th Working Conference on Mining Software Repositories. IEEE Press, 2013: 409-418.
[147]He P, Li B, Liu X, et al. An empirical study on software defect prediction with a simplified metric set[J]. Information and Software Technology, 2015, 59: 170-190.
[148]Madeyski L, Jureczko M. Which process metrics can significantly improve defect prediction models? An empirical study[J]. Software Quality Journal, 2015, 23(3): 393-422.
[149]Turhan B, Mısırlı A T, Bener A. Empirical evaluation of the effects of mixed project data on learning defect predictors[J]. Information and Software Technology, 2013, 55(6): 1101-1118.
[150]Chen L, Fang B, Shang Z, et al. Negative samples reduction in cross-company software defects prediction[J]. Information and Software Technology, 2015, 62: 67-77.
[151]Jureczko M. Significance of different software metrics in defect prediction[J]. Software Engineering: An International Journal, 2011, 1(1): 86-95.
[152]Bennin K E, Keung J, Phannachitta P, et al. Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction[J]. IEEE Transactions on Software Engineering, 2018, 44(6): 534-550.
[153]Ryu D, Jang J I, Baik J. A transfer cost-sensitive boosting approach for cross-project defect prediction[J]. Software Quality Journal, 2017, 25(1): 235-272.
[154]Shatnawi R. Deriving metrics thresholds using log transformation[J]. Journal of Software: Evolution and Process, 2015, 27(2): 95-113.
[155]Alenezi M, Magel K. Empirical evaluation of a new coupling metric: Combining structural and semantic coupling[J]. International Journal of Computers and Applications, 2014, 36(1): 34-44.
[156]Hosseini S, Turhan B, Mäntylä M. A benchmark study on the effectiveness of search-based data selection and feature selection for cross project defect prediction[J]. Information and Software Technology, 2018, 95: 296-312.
[157]Yu L. Using negative binomial regression analysis to predict software faults: A study of apache ant[J]. 2012.
[158]Mizuno O, Hirata Y. A cross-project evaluation of text-based fault-prone module prediction[C]//2014 6th International Workshop on Empirical Software Engineering in Practice. IEEE, 2014: 43-48.
[159]Kaur A, Kaur K. Performance analysis of ensemble learning for predicting defects in open source software[C]//2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2014: 219-225.
[160]Hosseini S, Turhan B, Mäntylä M. Search based training data selection for cross project defect prediction[C]//Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering. ACM, 2016: 3.
[161]Lumpe M, Vasa R, Menzies T, et al. Learning better inspection optimization policies[J]. International Journal of Software Engineering and Knowledge Engineering, 2012, 22(05): 621-644.
[162]Shatnawi R. The application of ROC analysis in threshold identification, data imbalance and metrics selection for software fault prediction[J]. Innovations in Systems and Software Engineering,

2017, 13(2-3): 201-217.

[163]Alenezi M, Zarour M. Modularity measurement and evolution in object-oriented open-source projects[C]//Proceedings of the The International Conference on Engineering & MIS 2015. ACM, 2015: 16.

[164]Poon W N, Bennin K E, Huang J, et al. Cross-project defect prediction using a credibility theory based naive bayes classifier[C]//2017 IEEE International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2017: 434-441.

[165]Xu Z, Liu J, Luo X, et al. Cross-version defect prediction via hybrid active learning with kernel principal component analysis[C]//2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2018: 209-220.

[166]Xu Z, Liu J, Luo X, et al. Software defect prediction based on kernel PCA and weighted extreme learning machine[J]. Information and Software Technology, 2019, 106: 182-200.

[167]Bennin K E, Keung J, Monden A. Impact of the distribution parameter of data sampling approaches on software defect prediction models[C]//2017 24th Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2017: 630-635.

[168]Bennin K E, Keung J W, Monden A. On the relative value of data resampling approaches for software defect prediction[J]. Empirical Software Engineering, 2019, 24(2): 602-636.

[169]Mizuno O. On effects of tokens in source code to accuracy of fault-prone module prediction[C]//2013 International Computer Science and Engineering Conference (ICSEC). IEEE, 2013: 103-108.

[170]Bluemke I, Stepień A. Experiment on defect prediction[C]//International Conference on Dependability and Complex Systems. Springer, Cham, 2015: 25-34.

[171]Gupta S, Gupta D L. Fault Prediction using Metric Threshold Value of Object Oriented Systems[J]. International Journal of Engineering Science, 2017, 13629.

[172]Shatnawi R. Synergies and conflicts among software quality attributes and bug fixes[J]. International Journal of Information Systems and Change Management, 2017, 9(1): 3-21.

[173]Yang Y. Software Defect Prediction Model Research for Network and Cloud Software Development[C]//2017 5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017). Atlantis Press, 2017.

[174]Shatnawi R. Improving Software Fault Prediction With Threshold Values[C]//2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, 2018: 1-6.

[175]Yuand Y T, Zhange T. Software Defect Prediction Based on Kernel PCA and Weighted Extreme Learning Machine[J].

[176]Bispo A, Prudêncio R, Véras D. Instance Selection and Class Balancing Techniques for Cross Project Defect Prediction[C]//2018 7th Brazilian Conference on Intelligent Systems (BRACIS). IEEE, 2018: 552-557.

[177]Ryu D, Baik J. A Comparative Study on Similarity Measure Techniques for Cross-Project Defect Prediction[J]. KIPS Transactions on Software and Data Engineering, 2018, 7(6): 205-220.

[178]Watanabe T, Monden A, Yücel Z, et al. Cross-validation-based association rule prioritization metric for software defect characterization[J]. IEICE TRANSACTIONS on Information and Systems, 2018, 101(9): 2269-2278.

[179]Malhotra R, Bansal A J. Fault prediction considering threshold effects of object-oriented metrics[J]. Expert Systems, 2015, 32(2): 203-219.

[180]Bennin K E, Keung J, Monden A, et al. The significant effects of data sampling approaches on software defect prioritization and classification[C]//Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE Press, 2017: 364-373.

[181]Alenezi M, Banitaan S, Obeidat Q. Fault-proneness of open source systems: An empirical analysis[J]. Synapse, 2014, 1: 256.

[182]Alenezi M, Zarour M. Does software structures quality improve over software evolution? Evidences from open-source projects[J]. International Journal of Computer Science and Information Security, 2016, 14: 61.

[183]He P, He Y, Yu L, et al. An Improved Method for Cross-Project Defect Prediction by Simplifying Training Data[J]. Mathematical Problems in Engineering, 2018, 2018.

[184]Jureczko M, Magott J. QualitySpy: a framework for monitoring software development processes[J]. Journal of Theoretical and Applied Computer Science, 2012, 6(1): 35-45.

[185]Jureczko M, Madeyski L. Cross–project defect prediction with respect to code ownership model: An empirical study[J]. e-Informatica Software Engineering Journal, 2015, 9(1).

[186]Shatnawi R. Empirical study of fault prediction for open-source systems using the Chidamber and Kemerer metrics[J]. IET software, 2013, 8(3): 113-119.

[187]Haouari A T, Souici-Meslati L, Atil F, et al. Empirical comparison and evaluation of Artificial Immune Systems in inter-release software fault prediction[J]. Applied Soft Computing, 2020: 106686.

[188]Tsoukalas D, Kehagias D, Siavvas M, et al. Technical debt forecasting: An empirical study on open-source repositories[J]. Journal of Systems and Software, 2020: 110777.

[189]Shao Y, Liu B, Wang S, et al. Software defect prediction based on correlation weighted class association rule mining[J]. Knowledge-Based Systems, 2020: 105742.

[190]Bal P R, Kumar S. WR-ELM: Weighted Regularization Extreme Learning Machine for Imbalance Learning in Software Fault Prediction[J]. IEEE Transactions on Reliability, 2020.

[191]Tong H, Liu B, Wang S, et al. Transfer-learning oriented class imbalance learning for cross-project defect prediction[J]. arXiv preprint arXiv:1901.08429, 2019.

[192]Alqmase M, Alshayeb M, Ghouti L. Threshold Extraction Framework for Software Metrics[J]. Journal of Computer Science and Technology, 2019, 34(5): 1063-1078.

[193]Xu Z, Li S, Xu J, et al. LDFR: Learning deep feature representation for software defect prediction[J]. Journal of Systems and Software, 2019, 158: 110402.

[194]Hosseini S. ASCIENTIAE RERUM[J].

[195]Sun Z, Li J, Sun H. An empirical study of public data quality problems in cross project defect prediction[J]. arXiv preprint arXiv:1805.10787, 2018.

[196]He P, Li B, Ma Y. Towards cross-project defect prediction with imbalanced feature sets[J]. arXiv preprint arXiv:1411.4228, 2014.

[197]Tantithamthavorn C, McIntosh S, Hassan A E, et al. Automated parameter optimization of classification techniques for defect prediction models[C]//Proceedings of the 38th International Conference on Software Engineering. 2016: 321-332.

[198]Tantithamthavorn C, McIntosh S, Hassan A E, et al. An empirical comparison of model validation techniques for defect prediction models[J]. IEEE Transactions on Software Engineering, 2016, 43(1): 1-18.

[199]Xia X, Lo D, Pan S J, et al. Hydra: Massively compositional model for cross-project defect prediction[J]. IEEE Transactions on software Engineering, 2016, 42(10): 977-998.

[200]Zhang F, Zheng Q, Zou Y, et al. Cross-project defect prediction using a connectivity-based unsupervised classifier[C]//2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE). IEEE, 2016: 309-320.

[201] Herbold S. Training data selection for cross-project defect prediction[C]//Proceedings of the 9th international conference on predictive models in software engineering. 2013: 1-10.
[202] Li J, He P, Zhu J, et al. Software defect prediction via convolutional neural network[C]//2017 IEEE International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2017: 318-328.
[203] He Z, Peters F, Menzies T, et al. Learning from open-source projects: An empirical study on defect prediction[C]//2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, 2013: 45-54.
[204] Herbold S, Trautsch A, Grabowski J. A comparative study to benchmark cross-project defect prediction approaches[J]. IEEE Transactions on Software Engineering, 2017, 44(9): 811-833.
[205] Tantithamthavorn C, McIntosh S, Hassan A E, et al. The impact of automated parameter optimization on defect prediction models[J]. IEEE Transactions on Software Engineering, 2018, 45(7): 683-711.
[206] Hosseini S, Turhan B, Gunarathna D. A systematic literature review and meta-analysis on cross project defect prediction[J]. IEEE Transactions on Software Engineering, 2017, 45(2): 111-147.
[207] Peters F, Menzies T, Layman L. LACE2: Better privacy-preserving data sharing for cross project defect prediction[C]//2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, 2015, 1: 801-811.
[208] Bowes D, Hall T, Petrić J. Software defect prediction: do different classifiers find the same defects?[J]. Software Quality Journal, 2018, 26(2): 525-552.
[209] Krishna R, Menzies T, Fu W. Too much automation? The bellwether effect and its implications for transfer learning[C]//Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. 2016: 122-131.
[210] Ryu D, Baik J. Effective multi-objective naïve Bayes learning for cross-project defect prediction[J]. Applied Soft Computing, 2016, 49: 1062-1077.
[211] Palomba F, Zanoni M, Fontana F A, et al. Toward a smell-aware bug prediction model[J]. IEEE Transactions on Software Engineering, 2017, 45(2): 194-218.
[212] Arar Ö F, Ayan K. Deriving thresholds of software metrics to predict faults on open source software: Replicated case studies[J]. Expert Systems with Applications, 2016, 61: 106-121.
[213] Herbold S, Trautsch A, Grabowski J. Global vs. local models for cross-project defect prediction[J]. Empirical software engineering, 2017, 22(4): 1866-1902.
[214] Zhou Y, Yang Y, Lu H, et al. How far we have progressed in the journey? an examination of cross-project defect prediction[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2018, 27(1): 1-51.
[215] Krishna R, Menzies T. Bellwethers: A baseline method for transfer learning[J]. IEEE Transactions on Software Engineering, 2018, 45(11): 1081-1105.
[216] Kawata K, Amasaki S, Yokogawa T. Improving relevancy filter methods for cross-project defect prediction[C]//2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence. IEEE, 2015: 2-7.
[217] Ma W, Chen L, Yang Y, et al. Empirical analysis of network measures for effort-aware fault-proneness prediction[J]. Information and Software Technology, 2016, 69: 50-70.
[218] Zhang F, Keivanloo I, Zou Y. Data transformation in cross-project defect prediction[J]. Empirical Software Engineering, 2017, 22(6): 3186-3218.
[219] Amasaki S, Kawata K, Yokogawa T. Improving cross-project defect prediction methods with data simplification[C]//2015 41st Euromicro Conference on Software Engineering and Advanced Applications. IEEE, 2015: 96-103.
[220] Yan M, Fang Y, Lo D, et al. File-level defect prediction: Unsupervised vs. supervised models[C]//2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). IEEE, 2017: 344-353.
[221] Turabieh H, Mafarja M, Li X. Iterated feature selection algorithms with layered recurrent neural network for software fault prediction[J]. Expert systems with applications, 2019, 122: 27-42.
[222] Boucher A, Badri M. Software metrics thresholds calculation techniques to predict fault-proneness: An empirical comparison[J]. Information and Software Technology, 2018, 96: 38-67.
[223] Huang J, Keung J W, Sarro F, et al. Cross-validation based K nearest neighbor imputation for software quality datasets: An empirical study[J]. Journal of Systems and Software, 2017, 132: 226-252.
[224] Li Z, Jing X Y, Zhu X, et al. Heterogeneous defect prediction through multiple kernel learning and ensemble learning[C]//2017 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2017: 91-102.
[225] Liu C, Yang D, Xia X, et al. A two-phase transfer learning model for cross-project defect prediction[J]. Information and Software Technology, 2019, 107: 125-136.
[226] Chen X, Zhang D, Zhao Y, et al. Software defect number prediction: Unsupervised vs supervised methods[J]. Information and Software Technology, 2019, 106: 161-181.
[227] Krishna R, Menzies T, Layman L. Less is more: Minimizing code reorganization using XTREE[J]. Information and Software Technology, 2017, 88: 53-66.
[228] Chen X, Shen Y, Cui Z, et al. Applying feature selection to software defect prediction using multi-objective optimization[C]//2017 IEEE 41st annual computer software and applications conference (COMPSAC). IEEE, 2017, 2: 54-59.
[229] Herbold S. CrossPare: A tool for benchmarking cross-project defect predictions[C]//2015 30th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW). IEEE, 2015: 90-96.
[230] Boucher A, Badri M. Using software metrics thresholds to predict fault-prone classes in object-oriented software[C]//2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD). IEEE, 2016: 169-176.
[231] Xiaoxing Yang, Wushao Wen. Ridge and Lasso Regression Models for Cross-Version Defect Prediction. IEEE Trans. Reliab. 67(3): 885-896 (2018).
[232] Zhang Y, Lo D, Xia X, et al. Combined classifier for cross-project defect prediction: an extended empirical study[J]. Frontiers of Computer Science, 2018, 12(2): 280-296.
[233] Herbold S. A systematic mapping study on cross-project defect prediction[J]. arXiv preprint arXiv:1705.06429, 2017.
[234] He P, Li B, Zhang D, et al. Simplification of training data for cross-project defect prediction[J]. arXiv preprint arXiv:1405.0773, 2014.
[235] Li Y, Huang Z, Wang Y, et al. Evaluating data filter on cross-project defect prediction: Comparison and improvements[J]. IEEE Access, 2017, 5: 25646-25656.
[236] Zhou Xu, Shuai Li, Yutian Tang, Xiapu Luo, Tao Zhang, Jin Liu, Jun Xu. Cross version defect prediction with representative data via sparse subset selection. ICPC 2018: 132-143.
[237] Ni C, Chen X, Wu F, et al. An empirical study on pareto based multi-objective feature selection for software defect prediction[J]. Journal of Systems and Software, 2019, 152: 215-238.

[238]Kondo M, Bezemer C P, Kamei Y, et al. The impact of feature reduction techniques on defect prediction models[J]. Empirical Software Engineering, 2019, 24(4): 1925-1963.
[239]Hussain S, Keung J, Khan A A, et al. Performance evaluation of ensemble methods for software fault prediction: An experiment[C]//Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference. 2015: 91-95.
[240]Boucher A, Badri M. Predicting fault-prone classes in object-oriented software: an adaptation of an unsupervised hybrid SOM algorithm[C]//2017 IEEE International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2017: 306-317.
[241]Hussain S, Keung J, Khan A A, et al. Detection of fault-prone classes using logistic regression based object-oriented metrics thresholds[C]//2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2016: 93-100.
[242]Porto F, Minku L, Mendes E, et al. A systematic study of cross-project defect prediction with meta-learning[J]. arXiv preprint arXiv:1802.06025, 2018.
[243]Zhang X, Ben K, Zeng J. Cross-entropy: A new metric for software defect prediction[C]//2018 IEEE International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2018: 111-122.
[244]Herbold S. Comments on ScottKnottESD in response to" An empirical comparison of model validation techniques for defect prediction models"[J]. IEEE Transactions on Software Engineering, 2017, 43(11): 1091-1094.
[245]Chen D, Fu W, Krishna R, et al. Applications of psychological science for actionable analytics[C]//Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2018: 456-467.
[246]Tang H, Lan T, Hao D, et al. Enhancing defect prediction with static defect analysis[C]//Proceedings of the 7th Asia-Pacific Symposium on Internetware. 2015: 43-51.
[247]Li Z, Jing X Y, Zhu X. Heterogeneous fault prediction with cost-sensitive domain adaptation[J]. Software Testing, Verification and Reliability, 2018, 28(2): e1658.
[248]Liu Y, Li Y, Guo J, et al. Connecting software metrics across versions to predict defects[C]//2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2018: 232-243.
[249]Qiu S, Lu L, Jiang S. Multiple-components weights model for cross-project software defect prediction[J]. IET Software, 2018, 12(4): 345-355.
[250]Yang J, Qian H. Defect prediction on unlabeled datasets by using unsupervised clustering[C]//2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2016: 465-472.
[251]Fan G, Diao X, Yu H, et al. Software Defect Prediction via Attention-Based Recurrent Neural Network[J]. Scientific Programming, 2019, 2019.
[252]Zhou T, Sun X, Xia X, et al. Improving defect prediction with deep forest[J]. Information and Software Technology, 2019, 114: 204-216.
[253]Prateek S, Pasala A, Aracena L M. Evaluating performance of network metrics for bug prediction in software[C]//2013 20th Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2013, 1: 124-131.
[254]Sousuke Amasaki. On Applicability of Cross-project Defect Prediction Method for Multi-Versions Projects. PROMISE 2017: 93-96.
[255]Morasca S, Lavazza L. Risk-averse slope-based thresholds: Definition and empirical evaluation[J]. Information and Software Technology, 2017, 89: 37-63.
[256]Mutlu B, Sezer E A, Akcayol M A. End-to-End hierarchical fuzzy inference solution[C]//2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2018: 1-9.
[257]Sousuke Amasaki. Cross-Version Defect Prediction using Cross-Project Defect Prediction Approaches: Does it work? PROMISE 2018: 32-41.
[258]Goyal R, Chandra P, Singh Y. Fuzzy inferencing to identify degree of interaction in the development of fault prediction models[J]. Journal of King Saud University-Computer and Information Sciences, 2017, 29(1): 93-102.
[259]Chen J, Hu K, Yang Y, et al. Collective transfer learning for defect prediction[J]. Neurocomputing, 2019.
[260]Li Z, Jing X Y, Zhu X, et al. Heterogeneous defect prediction with two-stage ensemble learning[J]. Automated Software Engineering, 2019, 26(3): 599-651.
[261]Porto F R, Simao A. Feature Subset Selection and Instance Filtering for Cross-project Defect Prediction-Classification and Ranking[J]. CLEI Electron. J., 2016, 19(3): 4.
[262]Herbold S. Benchmarking cross-project defect prediction approaches with costs metrics[J]. arXiv preprint arXiv:1801.04107, 2018.
[263]Pan C, Lu M, Xu B, et al. An Improved CNN Model for Within-Project Software Defect Prediction[J]. Applied Sciences, 2019, 9(10): 2138.
[264]Wen W, Zhang B, Gu X, et al. An empirical study on combining source selection and transfer learning for cross-project defect prediction[C]//2019 IEEE 1st International Workshop on Intelligent Bug Fixing (IBF). IEEE, 2019: 29-38.
[265]Catolino G, Palomba F, Fontana F A, et al. Improving change prediction models with code smell-related information[J]. Empirical Software Engineering, 2020, 25(1): 49-95.
[266]Cheikhi L, Abran A. An Analysis of the PROMISE and ISBSG Software Engineering Data Repositories[J]. Int. Journal of Computers and Technology, 2014, 13(5).
[267]Shriram C K, Muthukumaran K, Bhanu Murthy N L. Empirical study on the distribution of bugs in software systems[J]. International Journal of Software Engineering and Knowledge Engineering, 2018, 28(01): 97-122.
[268]Zhang X, Ben K, Zeng J. Using Cross-Entropy Value of Code for Better Defect Prediction[J]. International Journal of Performability Engineering, 2018, 14(9).
[269]Alenezi M, Abunadi I. Quality of open source systems from product metrics perspective[J]. arXiv preprint arXiv:1511.03194, 2015.
[270]Sohan M F, Kabir M A, Jabiullah M I, et al. Revisiting the class imbalance issue in software defect prediction[C]//2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2019: 1-6.
[271]Li K, Xiang Z, Chen T, et al. Understanding the Automated Parameter Optimization on Transfer Learning for CPDP: An Empirical Study[J]. arXiv preprint arXiv:2002.03148, 2020.
[272]Yucalar F, Ozcift A, Borandag E, et al. Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability[J]. Engineering Science and Technology, an International Journal, 2020, 23(4): 938-950.
[273]Chen D, Chen X, Li H, et al. Deepcpdp: Deep learning based cross-project defect prediction[J]. IEEE Access, 2019, 7: 184832-184848.
[274]Krishna R, Menzies T. Actionable= Cluster+ Contrast?[C]//2015 30th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW). IEEE, 2015: 14-17.

[275] Hosseini S, Turhan B. An exploratory study of search based training data selection for cross project defect prediction[C]//2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, 2018: 244-251.

[276] Muthukumaran K, Dasgupta A, Abhidnya S, et al. On the effectiveness of cost sensitive neural networks for software defect prediction[C]//International Conference on Soft Computing and Pattern Recognition. Springer, Cham, 2016: 557-570.

[277] Zhang D, Chen X, Cui Z, et al. Software defect prediction model sharing under differential privacy[C]//2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). IEEE, 2018: 1547-1554.

[278] Chen X, Mu Y, Qu Y, et al. Do different cross-project defect prediction methods identify the same defective modules?[J]. Journal of Software: Evolution and Process, 2020, 32(5): e2234.

[279] Kaur I, Bajpai N. An Empirical Study on Fault Prediction using Token-Based Approach[C]//Proceedings of the International Conference on Advances in Information Communication Technology & Computing. 2016: 1-7.

[280] Huo X, Yang Y, Li M, et al. Learning Semantic Features for Software Defect Prediction by Code Comments Embedding[C]//2018 IEEE International Conference on Data Mining (ICDM). IEEE, 2018: 1049-1054.

[281] Ghosh S, Rana A, Kansal V. A Hybrid Nonlinear Manifold Detection Approach for Software Defect Prediction[C]//2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, 2018: 453-459.

[282] Rosli M M, Tempero E, Luxton-Reilly A. What is in our datasets? Describing a structure of datasets[C]//Proceedings of the Australasian Computer Science Week Multiconference. 2016: 1-10.

[283] Wu Y, Huang S, Ji H. An Information Flow-based Feature Selection Method for Cross-Project Defect Prediction[J]. International Journal of Performability Engineering, 2018, 14(6).

[284] Di Nucci D, De Lucia A. The role of meta-learners in the adaptive selection of classifiers[C]//2018 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE). IEEE, 2018: 7-12.

[285] Tumar I, Hassouneh Y, Turabieh H, et al. Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction[J]. IEEE Access, 2020, 8: 8041-8055.

[286] Malhotra R. An extensive analysis of search-based techniques for predicting defective classes[J]. Computers & Electrical Engineering, 2018, 71: 611-626.

[287] Chen X, Zhang D, Cui Z Q, et al. Dp-share: Privacy-preserving software defect prediction model sharing through differential privacy[J]. Journal of Computer Science and Technology, 2019, 34(5): 1020-1038.

[288] Sohan M F, Jabiullah M I, Rahman S S M M, et al. Assessing the Effect of Imbalanced Learning on Cross-project Software Defect Prediction[C]//2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2019: 1-6.

[289] Sundström A. Investigation into predicting unit test failure using syntactic source code features[J]. 2018.

[290] Aziz S R, Khan T, Nadeem A. Experimental Validation of Inheritance Metrics' Impact on Software Fault Prediction[J]. IEEE Access, 2019, 7: 85262-85275.

[291] Ghosh S, Rana A, Kansal V. Evaluating the Impact of Sampling-Based Nonlinear Manifold Detection Model on Software Defect Prediction Problem[M]//Smart Intelligent Computing and Applications. Springer, Singapore, 2020: 141-152.

[292] Krishna R, Menzies T. Learning actionable analytics from multiple software projects[J]. Empirical Software Engineering, 2020: 1-33.

[293] Yang Y, Yang J, Qian H. Defect prediction by using cluster ensembles[C]//2018 Tenth International Conference on Advanced Computational Intelligence (ICACI). IEEE, 2018: 631-636.

[294] Jiarpakdee J, Tantithamthavorn C, Treude C. The impact of automated feature selection techniques on the interpretation of defect models[J]. Empirical Software Engineering, 2020: 1-49.

[295] Raukas H. Some approaches for software defect prediction[D]. Bachelor's Thesis (9ECTS), Institute of Computer Science Computer Science Curriculum, University of TARTU, 2017.

[296] Kaur K. Statistical Comparison of Machine Learning Techniques for Predicting Software Maintainability and Defects[J]. Guru Gobind Singh Indraprastha University, 2016.

[297] Okutan A. Use of Source Code Similarity Metrics in Software Defect Prediction[J]. arXiv preprint arXiv:1808.10033, 2018.

[298] Rizwan M, Nadeem A, Sindhu M A. Empirical Evaluation of Coupling Metrics in Software Fault Prediction[C]//2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST). IEEE, 2020: 434-440.

[299] Li K, Xiang Z, Chen T, et al. BiLO-CPDP: Bi-Level Programming for Automated Model Discovery in Cross-Project Defect Prediction[J]. arXiv preprint arXiv:2008.13489, 2020.

[300] Chen X, Mu Y, Ni C, et al. Revisiting Heterogeneous Defect Prediction: How Far Are We?[J]. arXiv preprint arXiv:1908.06560, 2019.

[301] Peng K, Menzies T. How to Improve AI Tools (by Adding in SE Knowledge): Experiments with the TimeLIME Defect Reduction Tool[J]. arXiv preprint arXiv:2003.06887, 2020.

[302] Yuan Z, Chen X, Cui Z, et al. ALTRA: Cross-Project Software Defect Prediction via Active Learning and Tradaboost[J]. IEEE Access, 2020, 8: 30037-30049.

[303] Li K, Xiang Z, Chen T, et al. Understanding the Automated Parameter Optimization on Transfer Learning for Cross-Project Defect Prediction: An Empirical Study[J].

[304] Sohan M F, Kabir M A, Rahman M, et al. Prevalence of Machine Learning Techniques in Software Defect Prediction[C]//International Conference on Cyber Security and Computer Science. Springer, Cham, 2020: 257-269.

[305] Krishnaa R, Menziesa T, Laymanb L. Recommendations for Intelligent Code Reorganization[J]. CoRR, 2016.

[306] Webster A. A Comparison of Transfer Learning Algorithms for Defect and Vulnerability Detection[R]. 2017.

[307] Amasaki S. Cross-version defect prediction: use historical data, cross-project data, or both?[J]. Empirical Software Engineering, 2020: 1-23.

[308] Deng J, Lu L, Qiu S, et al. A Suitable AST Node Granularity and Multi-Kernel Transfer Convolutional Neural Network for Cross-Project Defect Prediction[J]. IEEE Access, 2020, 8: 66647-66661.

[309] Ouellet A, Badri M. Empirical Analysis of Object-Oriented Metrics and Centrality Measures for Predicting Fault-Prone Classes in Object-Oriented Software[C]//International Conference on the Quality of Information and Communications Technology. Springer, Cham, 2019: 129-143.

[310] Qu Y, Liu T, Chi J, et al. node2defect: using network embedding to improve software defect prediction[C]//2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2018: 844-849.

[311] Ha D A, Chen T H, Yuan S M. Unsupervised methods for Software Defect Prediction[C]//Proceedings of the Tenth Interna-

tional Symposium on Information and Communication Technology. 2019: 49-55.
[312]Moudache S, Badri M. Software Fault Prediction Based on Fault Probability and Impact[C]//2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, 2019: 1178-1185.
[313]Ren J H, Liu F. Predicting Software Defects Using Self-Organizing Data Mining[J]. IEEE Access, 2019, 7: 122796-122810.
[314]Ahmed M R, Ali M A, Ahmed N, et al. The Impact of Software Fault Prediction in Real-World Application: An Automated Approach for Software Engineering[C]//Proceedings of 2020 the 6th International Conference on Computing and Data Engineering. 2020: 247-251.
[315]Bangash A A, Sahar H, Hindle A, et al. On the Time-Based Conclusion Stability of Software Defect Prediction Models[J]. arXiv preprint arXiv: 1911.06348v3, 2019.
[316]Sohan M F, Kabir M A, Rahman M, et al. Training Data Selection Using Ensemble Dataset Approach for Software Defect Prediction[C]//International Conference on Cyber Security and Computer Science. Springer, Cham, 2020: 243-256.
[317]Krishna Prasad R. Learning Actionable Analytics in Software Engineering[J]. 2019.
[318]Peng K, Menzies T. Defect Reduction Planning (using TimeLIME)[J]. arXiv preprint arXiv:2006.07416, 2020.
[319]Kumar S, Rathore S S. Evaluation of Techniques for Binary Class Classification[M]//Software Fault Prediction. Springer, Singapore, 2018: 39-57.
[320]Krishna R, Menzies T. From Prediction to Planning: Improving Software Quality with BELLTREE[J].
[321]Li H, Li X, Chen X, et al. Cross-project Defect Prediction via ASTToken2Vec and BLSTM-based Neural Network[C]//2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019: 1-8.
[322]Zhang J, Wu J, Chen C, et al. CDS: A Cross–Version Software Defect Prediction Model With Data Selection[J]. IEEE Access, 2020, 8: 110059-110072.
[323]Kumar K, Jayadev Gyani D, Narsimha G. Software Defect Prediction using Ant Colony Optimization[J]. International Journal of Applied Engineering Research, 2018, 13(19): 14291-14297.
[324]Morasca S, Lavazza L. On the assessment of software defect prediction models via ROC curves[J]. Empirical Software Engineering, 2020: 1-43.
[325]Zhang Q, Wu B. Software Defect Prediction via Transformer[C]//2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2020, 1: 874-879.
[326]Hussain S, Khan A A, Bennin K E. Empirical investigation of fault predictors in context of class membership probability estimation[C]//Proceedings of the 31st Annual ACM Symposium on Applied Computing. 2016: 1550-1553.
[327]Gao H, Lu M, Pan C, et al. Empirical Study: Are Complex Network Features Suitable for Cross-Version Software Defect Prediction?[C]//2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2019: 1-5.
[328]Mutlu B, Sezer E A, Akcayol M A. Automatic Rule Generation of Fuzzy Systems: A Comparative Assessment on Software Defect Prediction[C]//2018 3rd International Conference on Computer Science and Engineering (UBMK). IEEE, 2018: 209-214.
[329]De Lucia A, Salza P. Parallel Genetic Algorithms in the Cloud[J]. 2017.
[330]Catal C, Erdogan M, Isik C. Software Defect Prediction in the Cloud[J].
[331]Srivastava K. Cohesion Based Testability Model: A New Perspective[J].
[332]Fujiwara T, Mizuno O, Leelaprute P. Fault-Prone Byte-Code Detection Using Text Classifier[C]//International Conference on Product-Focused Software Process Improvement. Springer, Cham, 2015: 415-430.
[333]Fan G, Diao X, Yu H, et al. Deep Semantic Feature Learning with Embedded Static Metrics for Software Defect Prediction[C]//2019 26th Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2019: 244-251.
[334]AGRAWAL A. CROSS PROJECT DEFECT PREDICTION[D]. , 2018.
[335]Sara E, Laila C, Ali I. The Impact of SMOTE and Grid Search on Maintainability Prediction Models[C]//2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA). IEEE, 2019: 1-8.
[336]Chen H, Jing X Y, Li Z, et al. An Empirical Study on Heterogeneous Defect Prediction Approaches[J]. IEEE Transactions on Software Engineering, 2020.
[337]Ryu D, Baik J. Effective Harmony Search-Based Optimization of Cost-Sensitive Boosting for Improving the Performance of Cross-Project Defect Prediction[J]. KIPS Transactions on Software and Data Engineering, 2018, 7(3): 77-90.
[338]Rai P, Kumar S, Verma D K. Prediction of Software Effort Using Design Metrics: An Empirical Investigation[M]//Social Networking and Computational Intelligence. Springer, Singapore, 2020: 627-637.
[339]Bangash A A, Sahar H, Hindle A, et al. On the time-based conclusion stability of cross-project defect prediction models[J]. Empirical Software Engineering, 2020: 1-38.
[340]Yao Z, Song J, Liu Y, et al. Research on Cross-version Software Defect Prediction Based on Evolutionary Information[C]//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2019, 563(5): 052092.
[341]Kabir M A, Keung J W, Bennin K E, et al. A Drift Propensity Detection Technique to Improve the Performance for Cross-Version Software Defect Prediction[C]//2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2020: 882-891.
[342]Muthukumaran K, Murthy N L B, Janani P S. Empirical Study on the Distribution of Object-Oriented Metrics in Software Systems[C]//International Conference on Information and Software Technologies. Springer, Cham, 2019: 299-317.
[343]Peters F. LACE: Supporting Privacy-Preserving Data Sharing in Transfer Defect Learning[J]. 2014.
[344]Li J, Jing X Y, Wu F, et al. A Cost-Sensitive Shared Hidden Layer Autoencoder for Cross-Project Defect Prediction[C]//Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, Cham, 2019: 491-502.
[345]Di Nucci D. Methods and tools for focusing and prioritizing the testing effort[C]//2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2018: 722-726.
[346]Sun Y, Jing X Y, Wu F, et al. Adversarial Learning for Cross-Project Semi-Supervised Defect Prediction[J]. IEEE Access, 2020, 8: 32674-32687.
[347]Rathore S S, Kumar S. Homogeneous Ensemble Methods for the Prediction of Number of Faults[M]//Fault Prediction Modeling for the Prediction of Number of Software Faults. Springer, Singapore, 2019: 31-45.
[348]Tahir A, Bennin K E, MacDonell S G, et al. Revisiting the size effect in software fault prediction models[C]//Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. 2018: 1-10.
[349]Gong L, Jiang S, Jiang L. Conditional Domain Adversarial Adaptation for Heterogeneous Defect Prediction[J]. IEEE Access, 2020, 8: 150738-150749.

[350]Jiang K, Zhang Y, Wu H, et al. Heterogeneous defect prediction based on transfer learning to handle extreme imbalance[J]. Applied Sciences, 2020, 10(1): 396.

[351]Kaen E, Algarni A. Feature Selection Approach for Improving the Accuracy of Software Bug Prediction[J].

[352]Huo X, Li M. On cost-effective software defect prediction: Classification or ranking?[J]. Neurocomputing, 2019, 363: 339-350.

[353]Ren J, Liu F. A Novel Approach for Software Defect prediction Based on the Power Law Function[J]. Applied Sciences, 2020, 10(5): 1892.

[354]Maruf O M. The impact of parameter optimization of ensemble learning on defect prediction[J]. Computer Science Journal of Moldova, 2019, 79(1): 85-128.

[355]Xiaoxing Yang, Xin Li, Wushao Wen, Jianmin Su. An Investigation of Ensemble Approaches to Cross-Version Defect Prediction. SEKE 2019: 437-556.

[356]Peters F, Menzies T, Gong L, et al. Balancing privacy and utility in cross-company defect prediction[J]. IEEE Transactions on Software Engineering, 2013, 39(8): 1054-1068.

[357]Sarro F, Di Martino S, Ferrucci F, et al. A further analysis on the use of genetic algorithm to configure support vector machines for inter-release fault prediction[C]//Proceedings of the 27th annual ACM symposium on applied computing. 2012: 1215-1220.

[358]Rathore S S, Kumar S. Linear and non-linear heterogeneous ensemble methods to predict the number of faults in software systems[J]. Knowledge-Based Systems, 2017, 119: 232-256.

[359]Stuckman J, Walden J, Scandariato R. The effect of dimensionality reduction on software vulnerability prediction models[J]. IEEE Transactions on Reliability, 2016, 66(1): 17-37.

[360]Erturk E, Sezer E A. Iterative software fault prediction with a hybrid approach[J]. Applied Soft Computing, 2016, 49: 1020-1033.

[361]Foucault M, Falleri J R, Blanc X. Code ownership in open-source software[C]//Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. 2014: 1-9.

[362]Gupta D L, Saxena K. Software bug prediction using object-oriented metrics[J]. Sādhanā, 2017, 42(5): 655-669.

[363]Shukla S, Radhakrishnan T, Muthukumaran K, et al. Multi-objective cross-version defect prediction[J]. Soft Computing, 2018, 22(6): 1959-1980.

[364]Xuan X, Lo D, Xia X, et al. Evaluating defect prediction approaches using a massive set of metrics: An empirical study[C]//Proceedings of the 30th Annual ACM Symposium on Applied Computing. 2015: 1644-1647.

[365]Qing H, Biwen L, Beijun S, et al. Cross-project software defect prediction using feature-based transfer learning[C]//Proceedings of the 7th Asia-Pacific Symposium on Internetware. 2015: 74-82.

[366]Ferenc R, Tóth Z, Ladányi G, et al. A public unified bug dataset for Java[C]//Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering. 2018: 12-21.

[367]Kaur A, Kaur K. Value and applicability of academic projects defect datasets in cross-project software defect prediction[C]//2016 2nd International Conference on Computational Intelligence and Networks (CINE). IEEE, 2016: 154-159.

[368]Gong L, Jiang S, Bo L, et al. A novel class-imbalance learning approach for both within-project and cross-project defect prediction[J]. IEEE Transactions on Reliability, 2019, 69(1): 40-54.

[369]Wang F, Huang J, Ma Y. A Top-k Learning to Rank Approach to Cross-Project Software Defect Prediction[C]//2018 25th Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2018: 335-344.

[370]Du Y, Zhang L, Shi J, et al. Feature-grouping-based two steps feature selection algorithm in software defect prediction[C]//Proceedings of the 2nd International Conference on Advances in Image Processing. 2018: 173-178.

[371]Qiu S, Lu L, Cai Z, et al. Cross-Project Defect Prediction via Transferable Deep Learning-Generated and Handcrafted Features[C]//SEKE. 2019: 431-552.

[372]Goel L, Gupta S. Cross Projects Defect Prediction Modeling[M]//Data Visualization and Knowledge Engineering. Springer, Cham, 2020: 1-21.

[373]Cai Z, Lu L, Qiu S. An abstract syntax tree encoding method for cross-project defect prediction[J]. IEEE Access, 2019, 7: 170844-170853.

[374]Yu Q, Jiang S, Qian J, et al. Process metrics for software defect prediction in object-oriented programs[J]. IET Software, 2020.

[375]Geremia S, Tamburri D A. Varying defect prediction approaches during project evolution: A preliminary investigation[C]//2018 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE). IEEE, 2018: 1-6.

[376]Yang Z, Qian H. Automated Parameter Tuning of Artificial Neural Networks for Software Defect Prediction[C]//Proceedings of the 2nd International Conference on Advances in Image Processing. 2018: 203-209.

[377]Lu H. Semi-supervised and Active Learning Models for Software Fault Prediction[J]. 2015.

[378]Elahi E, Kanwal S, Asif A N. A new Ensemble approach for Software Fault Prediction[C]//2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST). IEEE, 2020: 407-412.

[379]Cui C, Liu B, Wang S. Isolation Forest Filter to Simplify Training Data for Cross-Project Defect Prediction[C]//2019 Prognostics and System Health Management Conference (PHM-Qingdao). IEEE, 2019: 1-6.

[380]Rosli M. A Framework for Understanding and Evaluating the Quality of Data Sets in Empirical Software Engineering[D]. ResearchSpace@ Auckland, 2018.

[381]Kumar L, Sureka A. Analyzing fault prediction usefulness from cost perspective using source code metrics[C]//2017 Tenth International Conference on Contemporary Computing (IC3). IEEE, 2017: 1-7.

[382]Choeikiwong T, Vateekul P. Two Stage Model to Detect and Rank Software Defects on Imbalanced and Scarcity Data Sets[J]. IAENG International Journal of Computer Science, 2016, 43(3).

[383]Xu Z, Li L, Yan M, et al. A comprehensive comparative study of clustering-based unsupervised defect prediction models[J]. Journal of Systems and Software, 2021, 172: 110862.

[384]Sun Z, Li J, Sun H, et al. CFPS: Collaborative filtering based source projects selection for cross-project defect prediction[J]. Applied Soft Computing, 2021, 99: 106940.

[385]Esteves G, Figueiredo E, Veloso A, et al. Understanding machine learning software defect predictions[J]. Automated Software Engineering, 2020, 27(3): 369-392.

[386]Shatnawi R. Comparison of threshold identification techniques for object-oriented software metrics[J]. IET Software, 2020, 14(6): 727-738.

[387]Mohamed F A, Salama C R, Yousef A H, et al. A Universal Model for Defective Classes Prediction Using Different Object-Oriented Metrics Suites[C]//2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES). IEEE, 2020: 65-70.

[388]Jin C. Cross-project software defect prediction based on domain adaptation learning and optimization[J]. Expert Systems with Applications, 2021, 171: 114637.

[389]Ferenc R, Tóth Z, Ladányi G, et al. A public unified bug dataset

for java and its assessment regarding metrics and bug prediction[J]. Software Quality Journal, 2020: 1-60.
[390]Chen X, Mu Y, Liu K, et al. Revisiting heterogeneous defect prediction methods: How far are we?[J]. Information and Software Technology, 2021, 130: 106441.
[391]Pecorelli F, Di Nucci D. Adaptive selection of classifiers for bug prediction: A large-scale empirical analysis of its performances and a benchmark study[J]. Science of Computer Programming, 2021, 205: 102611.
[392]Eken B, Palma F, Ayşe B, et al. An empirical study on the effect of community smells on bug prediction[J]. Software Quality Journal, 2021, 29(1): 159-194.
[393]Alkhaeir T, Walter B. The effect of code smells on the relationship between design patterns and defects[J]. IEEE Access, 2020.
[394]Hassouneh Y, Turabieh H, Thaher T, et al. Boosted Whale Optimization Algorithm With Natural Selection Operators for Software Fault Prediction[J]. IEEE Access, 2021, 9: 14239-14258.
[395]Eivazpour Z, Keyvanpour M R. CSSG: A cost-sensitive stacked generalization approach for software defect prediction[J]. Software Testing, Verification and Reliability, 2021: e1761.
[396]Niu L, Wan J, Wang H, et al. Cost-sensitive Dictionary Learning for Software Defect Prediction[J]. Neural Processing Letters, 2020, 52(3): 2415-2449.
[397]Zhou X, Lu L. Defect Prediction via LSTM Based on Sequence and Tree Structure[C]//2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2020: 366-373.
[398]Zou Q, Lu L, Qiu S, et al. Correlation feature and instance weights transfer learning for cross project software defect prediction[J]. IET Software, 2021, 15(1): 55-74.
[399]Wang A, Zhang Y, Yan Y. Heterogeneous Defect Prediction Based on Federated Transfer Learning via Knowledge Distillation[J]. IEEE Access, 2021, 9: 29530-29540.
[400]Malhotra R, Jain J. Predicting Software Defects for Object-Oriented Software Using Search-based Techniques[J]. International Journal of Software Engineering and Knowledge Engineering, 2021, 31(02): 193-215.
[401]Li X, Yang X, Su J, et al. A Multi-Objective Learning Method for Building Sparse Defect Prediction Models[C]//2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS). IEEE, 2020: 204-211.
[402]Lavazza L, Morasca S. An Empirical Study of Thresholds for Code Measures[C]//2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2020: 346-357.
[403]Jahanshahi H, Cevik M, Başar A. Moving from cross-project defect prediction to heterogeneous defect prediction: a partial replication study[J]. arXiv preprint arXiv:2103.03490, 2021.
[404]Malhotra R, Jain J. Predicting defects in object-oriented software using cost-sensitive classification[C]//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2021, 1022(1): 012112.
[405]Wu Y, Yao J, Chang S, et al. LIMCR: Less-Informative Majorities Cleaning Rule Based on Naïve Bayes for Imbalance Learning in Software Defect Prediction[J]. Applied Sciences, 2020, 10(23): 8324.
[406]Wang H, Zhuang W, Zhang X. Software Defect Prediction Based on Gated Hierarchical LSTMs[J]. IEEE Transactions on Reliability, 2021.
[407]Yadav H S. Increasing Accuracy of Software Defect Prediction using 1-dimensional CNN with SVM[C]//2020 IEEE International Conference for Innovation in Technology (INOCON). IEEE, 2020: 1-6.
[408]Yan X, Zhang Y, Khan A A. An algorithm acceleration framework for correlation-based feature selection[C]//MATEC Web of Conferences. EDP Sciences, 2021, 336: 07011.

The list of citations list using the JIRA-HA-2019 / JIRA-RA-2019 data set

[409]S. Wattanakriengkrai, P. Thongtanunam, C. Tantithamthavorn, H. Hata, K. Matsumoto. Predicting Defective Lines Using a Model-Agnostic Technique. arXiv preprint arXiv:2009.03612v1, 2020.
[410]Kushani Perera, Jeffrey Chan, Shanika Karunasekera: A Framework for Feature Selection to Exploit Feature Group Structures. PAKDD (1) 2020: 792-804.
[411]Rajapaksha D, Tantithamthavorn C, Jiarpakdee J, et al. SQAPlanner: Generating Data-InformedSoftware Quality Improvement Plans[J]. arXiv preprint arXiv:2102.09687, 2021.
[412]Jiarpakdee J, Tantithamthavorn C, Grundy J. Practitioners' Perceptions of the Goals and Visual Explanations of Defect Prediction Models[J]. arXiv preprint arXiv:2102.12007, 2021.
[413]Jahanshahi H, Cevik M, Başar A. Moving from cross-project defect prediction to heterogeneous defect prediction: a partial replication study[J]. arXiv preprint arXiv:2103.03490, 2021.
[414]Khan S S, Niloy N T, Azmain M A, et al. Impact of Label Noise and Efficacy of Noise Filters in Software Defect Prediction[J].

The list of citations list using the IND-JLMIV+R-2020 data set

[415]Alexander Trautsch, Fabian Trautsch, Steffen Herbold, Benjamin Ledel, Jens Grabowski. The SmartSHARK Ecosystem for Software Repository Mining. arXiv preprint arXiv:2001.01606v1, 2020.
[416]Alexander Trautsch, Steffen Herbold, Jens Grabowski. A Longitudinal Study of Static Analysis Warning Evolution and the Effects of PMD on Software Quality in Apache Open Source Projects. arXiv preprint arXiv:1912.02179v3, 2020.
[417]Steffen Herbold, Alexander Trautsch, Benjamin Ledel. Large-Scale Manual Validation of Bugfixing Changes. MSR 2020.
[418]Herbold, S., (2020). Autorank: A Python package for automated ranking of classifiers. Journal of Open Source Software, 5(48), 2173. https://doi.org/10.21105/joss.02173.
[419]Herbold S, Trautsch A, Trautsch F. On the feasibility of automated prediction of bug and non-bug issues[J]. Empirical Software Engineering, 2020: 1-37.