Literaturliste

Fehlervorhersage

Ceylan, E., Kutlubay, F. O., & Bener, A. B. (2006). Software defect identification using machine learning techniques. *Proceedings - 32nd Euromicro Conference on Software Engineering and Advanced Applications, SEAA*, 240–246. https://doi.org/10.1109/EUROMICRO.2006.56

Challagulla, V. U. B., Bastani, F. B., Yen, I. L., & Paul, R. A. (2008). Empirical assessment of machine learning based software defect prediction techniques. *International Journal on Artificial Intelligence Tools*, *17*(2), 389–400. https://doi.org/10.1142/S0218213008003947

Dhiauddin, M., & Ibrahim, S. (2012). A Prediction Model for System Testing Defects using Regression Analysis. *International Journal of Soft Computing and Software Engineering*, *2*(7), 55–68. https://doi.org/10.7321/jscse.v2.n7.6

Hammouri, A., Hammad, M., Alnabhan, M., & Alsarayrah, F. (2018). Software Bug Prediction using machine learning approach. *International Journal of Advanced Computer Science and Applications*, *9*(2), 78–83. https://doi.org/10.14569/IJACSA.2018.090212

Li, J., He, P., Zhu, J., & Lyu, M. R. (2017). Software defect prediction via convolutional neural network. *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security, QRS 2017*, 318–328. https://doi.org/10.1109/QRS.2017.42

Queiroz, R., Berger, T., & Czarnecki, K. (2016). Towards predicting feature defects in software product lines. *FOSD 2016 - Proceedings of the 7th International Workshop on Feature-Oriented Software Development, Co-Located with SPLASH 2016*, 58–62. https://doi.org/10.1145/3001867.3001874

Ratzinger, J., Sigmund, T., & Gall, H. C. (2008). On the relation of refactoring and software defects. *Proceedings - International Conference on Software Engineering*, 35–38. https://doi.org/10.1145/1370750.1370759

Son, L. H., Pritam, N., Khari, M., Kumar, R., Phuong, P. T. M., & Thong, P. H. (2019). Empirical study of software defect prediction: A systematic mapping. *Symmetry*, *11*(2). https://doi.org/10.3390/sym11020212

Song, Q., Jia, Z., Shepperd, M., Ying, S., & Liu, J. (2011). A General Software Defect-Proneness Prediction Framework. *IEEE Transactions on Software Engineering*, *37*(3), 356–370. https://doi.org/10.1109/TSE.2010.90

Zimmermann, T., Premraj, R., & Zeller, A. (2007). Predicting defects for eclipse. *Proceedings - ICSE 2007 Workshops: Third International Workshop on Predictor Models in Software Engineering, PROMISE'07*. https://doi.org/10.1109/PROMISE.2007.10

GitHub Mining

Joblin, M., Mauerer, W., Apel, S., Siegmund, J., & Riehle, D. (2015). From developer networks to verified communities: A fine-grained approach. *Proceedings - International Conference on Software Engineering*, *1*, 563–573. https://doi.org/10.1109/ICSE.2015.73

Kagdi, H., Collard, M. L., & Maletic, J. I. (2007). A survey and taxonomy of approaches for mining software repositories in the context of software evolution. *Journal of Software Maintenance and Evolution: Research and Practice*, *19*(2), 77–131. https://doi.org/10.1002/smr.344

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., & Damian, D. (2014). The promises and perils of mining GitHub. *Proceedings of the 11th Working Conference on Mining Software Repositories - MSR 2014*, *3109*, 92–101. https://doi.org/10.1145/2597073.2597074

Muthukumaran, K., Choudhary, A., & Murthy, N. L. B. (2015). Mining github for novel change metrics to predict buggy files in software systems. *Proceedings - 1st International Conference on Computational Intelligence and Networks, CINE 2015*, 15–20. https://doi.org/10.1109/CINE.2015.13

Russell, M. A., & Klassen, M. (2018). *Mining the Social Web - Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and more*. O'Reilly.

Tan, M., Tan, L., Dara, S., & Mayeux, C. (2015). Online Defect Prediction for Imbalanced Data. *Proceedings - International Conference on Software Engineering*, *2*, 99–108. https://doi.org/10.1109/ICSE.2015.139

Williams, C. C., & Hollingsworth, J. K. (2005). Automatic mining of source code repositories to improve bug finding techniques. *IEEE Transactions on Software Engineering*, *31*(6), 466–480. https://doi.org/10.1109/TSE.2005.63

Xie, T., & Pei, J. (2006). MAPO. *Proceedings of the 2006 International Workshop on Mining Software Repositories - MSR '06*, 54. https://doi.org/10.1145/1137983.1137997

Methodik

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0. *CRISP-DM Consortium*, 76. https://doi.org/10.1109/ICETET.2008.239

Metriken

Berger, T., & Guo, J. (2014). Towards system analysis with variability model metrics. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/2556624.2556641

Gao, K., Khoshgoftaar, T. M., Wang, H., & Seliya, N. (2011). Choosing software metrics for defect prediction: an investigation on feature selection techniques. *Software: Practice and Experience*, *41*(5), 579–606. https://doi.org/10.1002/spe.1043

Kaur, A., Kaur, K., & Kaur, H. (2015). An investigation of the accuracy of code and process metrics for defect prediction of mobile applications. *2015 4th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2015*, 1–6. https://doi.org/10.1109/ICRITO.2015.7359220

Krüger, J., Gu, W., Shen, H., Mukelabai, M., Hebig, R., & Berger, T. (2018). Towards a beter understanding of software features and their characteristics: A case study of Marlin. *ACM International Conference Proceeding Series*, 105–112. https://doi.org/10.1145/3168365.3168371

Lee, T., Nam, J., Han, D., Kim, S., & Peter In, H. (2016). Developer Micro Interaction Metrics for Software Defect Prediction. *IEEE Transactions on Software Engineering*, *42*(11), 1015–1035. https://doi.org/10.1109/TSE.2016.2550458

Madeyski, L., & Jureczko, M. (2015). Which process metrics can significantly improve defect prediction models? An empirical study. *Software Quality Journal*, *23*(3), 393–422. https://doi.org/10.1007/s11219-014-9241-7

Moser, R., Pedrycz, W., & Succi, G. (2008). A Comparative analysis of the efficiency of change metrics and static code attributes for defect prediction. *Proceedings - International Conference on Software Engineering*, 181–190. https://doi.org/10.1145/1368088.1368114

Nagappan, N., Ball, T., & Zeller, A. (2006). Mining metrics to predict component failures. *Proceedings - International Conference on Software Engineering*, *2006*, 452–461. https://doi.org/10.1145/1134285.1134349

Okutan, A., & Yıldız, O. T. (2014). Software defect prediction using Bayesian networks. *Empirical Software Engineering*, *19*(1), 154–181. https://doi.org/10.1007/s10664-012-9218-8

Rahman, F., & Devanbu, P. (2013). How, and why, process metrics are better. *Proceedings - International Conference on Software Engineering*, 432–441. https://doi.org/10.1109/ICSE.2013.6606589

Singh, G., Singh, D., & Singh, V. (2011). A study of software metrics. *IJCEM International Journal of Computational Engineering & Management*, *11*, 22–27.

Son, L. H., Pritam, N., Khari, M., Kumar, R., Phuong, P. T. M., & Thong, P. H. (2019). Empirical study of software defect prediction: A systematic mapping. *Symmetry*, *11*(2). https://doi.org/10.3390/sym11020212

Wang, H., Khoshgoftaar, T. M., & Seliya, N. (2011). How many software metrics should be selected for defect prediction? *Proceedings of the 24th International Florida Artificial Intelligence Research Society, FLAIRS - 24*, (Mi), 69–74.

Machine Learning

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *ACM International Conference Proceeding Series*, *148*, 161–168. https://doi.org/10.1145/1143844.1143865

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. In *Synthesis Lectures on Mathematics and Statistics*. https://doi.org/10.1007/978-1-4614-7138-7

Jukes, E. (2017). Encyclopedia of Machine Learning and Data Mining. In C. Sammut & G. I. Webb (Eds.), *Reference Reviews* (Vol. 32). https://doi.org/10.1007/978-1-4899-7687-1

Khoshgoftaar, T. M., Gao, K., & Seliya, N. (2010). Attribute selection and imbalanced data: Problems in software defect prediction. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *1*, 137–144. https://doi.org/10.1109/ICTAI.2010.27

Seliya, N., Khoshgoftaar, T. M., & Van Hulse, J. (2009). A study on the relationships of classifier performance metrics. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 59–66. https://doi.org/10.1109/ICTAI.2009.25

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. *AAAI Workshop - Technical Report*, *WS-06-06*, 24–29. https://doi.org/10.1007/11941439_114

SPL allgemein

Apel, S., Batory, D., Kästner, C., & Saake, G. (2013). *Feature-Oriented Software Product Lines*. https://doi.org/10.1007/978-3-642-37521-7

Berger, T., Lettner, D., Rubin, J., Grünbacher, P., Silva, A., Becker, M., … Czarnecki, K. (2015). What is a feature? *Proceedings of the 19th International Conference on Software Product Line - SPLC '15*, *3*(1), 16–25. https://doi.org/10.1145/2791060.2791108

Lee, K., Kang, K. C., & Lee, J. (2002). Concepts and guidelines of feature modeling for product line software engineering. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2319*(April), 62–77. https://doi.org/10.1007/3-540-46020-9_5

Liebig, J., Apel, S., Lengauer, C., Kästner, C., & Schulze, M. (2010). An analysis of the variability in forty preprocessor-based software product lines. *Proceedings - International Conference on Software Engineering*, *1*, 105–114. https://doi.org/10.1145/1806799.1806819

Pohl, K., Böckle, G., & van der Linden, F. (2005). *Software Product Line Engineering*. https://doi.org/10.1007/3-540-28901-1

Queiroz, R., Passos, L., Valente, M. T., Hunsen, C., Apel, S., & Czarnecki, K. (2017). The shape of feature code: an analysis of twenty C-preprocessor-based systems. *Software and Systems Modeling*, *16*(1), 77–96. https://doi.org/10.1007/s10270-015-0483-z

Sincero, J., Tartler, R., Lohmann, D., & Schröder-Preikschat, W. (2011). Efficient extraction and analysis of preprocessor-based variability. *ACM SIGPLAN Notices*, *46*(2), 33–42. https://doi.org/10.1145/1942788.1868300

Thüm, T., Apel, S., Kästner, C., Schaefer, I., & Saake, G. (2014). A classification and survey of analysis strategies for software product lines. *ACM Computing Surveys*, *47*(1). https://doi.org/10.1145/2580950

TM Programmcode

Charoenwet, W. (2018). A Digital collection study and framework exploration-Applying textual analysis on source code collection. *Proceedings of the 2018 3rd Digital Heritage International Congress, Digital Heritage 2018 - Held Jointly with the 2018 24th International Conference on Virtual Systems and Multimedia, VSMM 2018*, 1–8. https://doi.org/10.1109/DigitalHeritage.2018.8810105

Dreweke, A., Fischer, I., Werth, T., & Wörlein, M. (2010). Text Mining in Program Code. In *Handbook of Research on Text and Web Mining Technologies* (pp. 626–645). https://doi.org/10.4018/978-1-59904-990-8.ch035