Literaturliste

Aishwarya V Srinivasan. (2019). *Stochastic Gradient Descent — Clearly Explained !! - Towards Data Science*. Retrieved from https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31

Alam, M. S., & Vuong, S. T. (2013). Random forest classification for detecting android malware. *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-IThings-CPSCom 2013*, 663–669. https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.122

Alpaydin, E. (2010). *Introduction to Machine Learning* (Second Edi). Cambridge, Massachusetts: The MIT Press.

Alsaeedi, A., & Khan, M. Z. (2019). Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study. *Journal of Software Engineering and Applications*, *12*(05), 85–100. https://doi.org/10.4236/jsea.2019.125007

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

Borg, M., Svensson, O., Berg, K., & Hansson, D. (2019). *SZZ unleashed: an open implementation of the SZZ algorithm - featuring example usage in a study of just-in-time bug prediction for the Jenkins project*. 7–12. https://doi.org/10.1145/3340482.3342742

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

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

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

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

Diab, S. (2019). *Optimizing Stochastic Gradient Descent in Text Classification Based on Fine-Tuning Hyper-Parameters Approach. A Case Study on Automatic Classification of Global Terrorist Attacks*. *16*(12), 1–6. Retrieved from http://arxiv.org/abs/1902.06542

Frank, E., Hall, M. A., & Witten, I. H. (2016). *WEKA Workbench*.

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

Hunsen, C., Zhang, B., Siegmund, J., Kästner, C., Leßenich, O., Becker, M., & Apel, S. (2016). Preprocessor-based variability in open-source and industrial software systems: An empirical study. In *Empirical Software Engineering* (Vol. 21). https://doi.org/10.1007/s10664-015-9360-1

KNIMETV. (2019). *What is an ROC-Curve?* (Vol. 34). Vol. 34.

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

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

Linder, R., Geier, J., & Kölliker, M. (2004). Artificial neural networks, classification trees and regression: Which method for which customer base? *Journal of Database Marketing & Customer Strategy Management*, *11*(4), 344–356. https://doi.org/10.1057/palgrave.dbm.3240233

Luber, S., & Litzel, N. (2019). *Was ist eine Support Vector Machine?*

Medeiros, F., Ribeiro, M., Gheyi, R., Apel, S., Kästner, C., Ferreira, B., … Fonseca, B. (2018). Discipline Matters: Refactoring of Preprocessor Directives in the #ifdef Hell. *IEEE Transactions on Software Engineering*, *44*(5), 453–469. https://doi.org/10.1109/TSE.2017.2688333

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

O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks*. 1–11. Retrieved from http://arxiv.org/abs/1511.08458

Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *Journal of Educational Research*, *96*(1), 3–14. https://doi.org/10.1080/00220670209598786

Preschern, C. (2019). Patterns to escape the #ifdef hell. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3361149.3361151

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

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

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

Raschka, S. (2014). *Naive Bayes and Text Classification I - Introduction and Theory*. 1–20. Retrieved from http://arxiv.org/abs/1410.5329

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

Rohith, G. (2018). Support Vector Machine — Introduction to Machine Learning Algorithms. *2018*, p. 1. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Rokach, L., & Maimon, O. (2013). Decision Trees. In *Data Mining and Knowledge Discovery Handbook* (pp. 165–192). https://doi.org/10.1007/0-387-25465-X_9

Sammut, C., & Webb, G. I. (2017). *Encyclopedia of Machine Learning and Data Mining* (Second edi).

https://doi.org/10.1007/978-1-4899-7687-1

Sarang Narkhede. (2018). Understanding AUC - ROC Curve - Towards Data Science. *Towards Data Science*. Retrieved from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Śliwerski, J., Zimmermann, T., & Zeller, A. (2005). When do changes induce fixes? *Proceedings of the 2005 International Workshop on Mining Software Repositories, MSR 2005*. https://doi.org/10.1145/1082983.1083147

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

Spadini, D., Aniche, M., & Bacchelli, A. (2018). PyDriller: Python framework for mining software repositories. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering - ESEC/FSE 2018*, 908–911. https://doi.org/10.1145/3236024.3264598

Stallman, R. M., & Weinberg, Z. (1987). *The C Preprocessor*.

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

Wang, J., Shen, B., & Chen, Y. (2012). Compressed C4.5 models for software defect prediction. *Proceedings - International Conference on Quality Software*, *2*(1), 13–16. https://doi.org/10.1109/QSIC.2012.19

Wen, M., Wu, R., Liu, Y., Tian, Y., Xie, X., Cheung, S. C., & Su, Z. (2019). Exploring and exploiting the correlations between bug-inducing and bug-fixing commits. *ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 326–337. https://doi.org/10.1145/3338906.3338962

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, *4*(11), 1–7. https://doi.org/10.21037/atm.2016.03.37

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