REFERENCES

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and Regression Trees. CRC Press.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: a survey and categorization. Information Fusion, 6(1), 5-20.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem. In Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (pp. 475-482). Springer, Berlin, Heidelberg.

Catal, C., & Sezer, E. B. (2011). A systematic review of software fault prediction studies. Expert Systems with Applications, 38(2), 1350-1357.

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.

Chinnathambi, Arun & Lakshmi, C.. (2020). Class Imbalance in Software Fault Prediction Data Set. 10.1007/978-981-15-0199-9_64.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

D'Ambros, M., Lanza, M., & Robbes, R. (2010). An extensive comparison of bug prediction approaches. In 7th IEEE Working Conference on Mining Software Repositories (MSR 2010). IEEE.

Domingos, P., & Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Machine Learning, 29(2-3), 103-130.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. Journal of Artificial Intelligence Research, 61, 863-905.

Fix, E., & Hodges, J. L. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. USAF School of Aviation Medicine.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of statistical software, 33(1), 1.

Gao, Xianwei & Shan, Chun & Hu, Changzhen & Niu, Zequn & Liu, Zachary. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. IEEE Access. 7. 1-1. 10.1109/ACCESS.2019.2923640.

Ghotra, B., McIntosh, S., & Hassan, A. E. (2015). Revisiting the impact of classification techniques on the performance of defect prediction models. In Proceedings of the 37th International Conference on Software Engineering-Volume 1 (pp. 789-800). IEEE Press.

Gray, D., Bowes, D., Davey, N., Sun, Y., & Christianson, B. (2012). The misuse of the NASA metrics data program data sets for automated software defect prediction. In 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011) (Vol. 26, No. 1, pp. 96-103).

Hamoud Aljamaan and Amal Alazba. 2020. Software Defect Prediction using Tree-Based Ensembles. In Proceedings of the 16th ACM International Conference

on Predictive Models and Data Analytics in Software Engineering (PROMISE '20), November 8–9, 2020, Virtual, USA. ACM, New York, NY, USA, 10 pages.

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In International Conference on Intelligent Computing (pp. 878-887). Springer, Berlin, Heidelberg.

He, H., & Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 1322-1328). IEEE.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284.

Hongyan Wan, Guoqing Wu, Mali Yu, Mengting Yuan: Software Defect Prediction Based on Cost-Sensitive Dictionary Learning. Int. J. Softw. Eng. Knowl. Eng. 29(9): 1219-1243 (2019).

Izquierdo, J. L. A., & Segovia, J. (2009). Optimization of bagging ensembles via a genetic algorithm. In 2009 IEEE International Conference on Fuzzy Systems (pp. 2175-2180). IEEE.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

Jureczko, M., & Madeyski, L. (2010). Towards identifying software project clusters with regard to defect prediction. In Proceedings of the 6th International Conference on Predictive Models in Software Engineering.

Kumar, Vipin & Minz, Sonajharia. (2016). Multi-view ensemble learning: An optimal feature set partitioning for high-dimensional data classification. Knowledge and Information Systems. 2015.10.1007/s10115-015-0875-y.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine learning, 51(2), 181-207.

Leveson, N. G. (2011). Engineering a safer world: Systems thinking applied to safety. MIT press.

Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking classification models for software defect prediction: A proposed framework and novel findings. IEEE Transactions on Software Engineering, 34(4), 485-496.

Lingden, P. & Alsadoon, A. & P.W.C, Prasad & Alsadoon, Omar & Ali, Rasha & Tran Quoc Vinh, Nguyen. (2019). A novel modified undersampling (MUS) technique for software defect prediction. Computational Intelligence. 35. 10.1111/coin.12229.

McConnell, S. (2004). Code complete. Pearson Education.

Menzies, T., Dekhtyar, A., Distefano, J., & Greenwald, J. (2007). Problems with precision: A response to" comments on 'Data mining static code attributes to learn defect predictors'". IEEE Transactions on Software Engineering, 33(9), 637-640.

Polikar, R. (2006). Ensemble based systems in decision making. IEEE Circuits and systems magazine, 6(3), 21-45.

Polikar, R. (2012). Ensemble learning. In Ensemble machine learning (pp. 1-34). Springer.

Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81-106.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann.

Rasmussen, C. E., & Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. MIT Press.

Schölkopf, B., & Smola, A. J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Shepperd, M., Song, Q., Sun, Z., & Mair, C. (2013). Data quality: Some comments on the NASA software defect datasets. IEEE Transactions on Software Engineering, 39(9), 1208-1215.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Wahono, R. S. (2015). A systematic literature review of software defect prediction: Research trends, datasets, methods and frameworks. Journal of Software Engineering, 1(1), 1-16.

Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.

Yu, Zhiwen & Wang, Daxing & You, Jane & Wong, Hau-San & Wu, Si & Zhang, Jun & Han, Guoqiang. (2016). Progressive Subspace Ensemble Learning. Pattern Recognition. 60.10.1016/j.patcog.2016.06.017.

Z. Li, X. Zhang, J. Guo and Y. Shang, "Class Imbalance Data-Generation for Software Defect Prediction," 2019 26th Asia-Pacific Software Engineering

Conference (APSEC), Putrajaya, Malaysia, 2019, pp. 276-283, doi: 10.1109/APSEC48747.2019.00045

Zhang, H., & Ma, S. (2012). Ensemble Learning for Software Defect Prediction. In 34th International Conference on Software Engineering (ICSE), Zurich, Switzerland.

Zhou, Z.-H. (2012). Ensemble methods: foundations and algorithms. Chapman and Hall/CRC.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301-320.