

References

- [1] F.-M. Lazar and O. Baniyas, "Clone detection algorithm based on the Abstract Syntax Tree approach," in 2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2014, pp. 73–78. Available: <http://ieeexplore.ieee.org/document/6840038/>
- [2] H. A. Basit and S. Jarzabek, "A Data Mining Approach for Detecting Higher-Level Clones in Software," IEEE Transactions on Software Engineering, vol. 35, no. 4, pp. 497–514, Jul. 2009. Available: <http://ieeexplore.ieee.org/document/4796208/>
- [3] N. Göde and R. Koschke, "Incremental Clone Detection," in 2009 13th European Conference on Software Maintenance and Reengineering, Kaiserslautern, Germany, 2009, pp. 219–228. Available: <http://ieeexplore.ieee.org/document/4812755/>
- [4] F. Deissenboeck, B. Hummel, and E. Juergens, "Code clone detection in practice," in *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10*, Cape Town, South Africa, 2010, vol. 2, p. 499. Available: <https://ieeexplore.ieee.org/document/6062267/>
- [5] P. Gautam and H. Saini, "Various Code Clone Detection Techniques and Tools: A Comprehensive Survey," in *Smart Trends in Information Technology and Computer Communications*, vol. 628, A. Unal, M. Nayak, D. K. Mishra, D. Singh, and A. Joshi, Eds. Singapore: Springer Singapore, 2016, pp. 655–667. Available: http://link.springer.com/10.1007/978-981-10-3433-6_79
- [6] C. K. Roy and J. R. Cordy, "Scenario-Based Comparison of Clone Detection Techniques," in 2008 16th IEEE International Conference on Program Comprehension, Amsterdam, 2008, pp. 153–162. Available: <http://ieeexplore.ieee.org/document/4556127/>
- [7] K. Kaur, "A Comprehensive Review of Code Clone Detection Techniques," p. 5, 2015.
- [8] B. S. Baker, "On finding duplication and near-duplication in large software systems," in 2nd Working Conference on Reverse Engineering, 1995.

- [9] H. Kaur and R. Kaur, "Clone Detection in Web Application Using Clone Metrics," *International Journal of Advanced Research in Computer Science and Software Engineering*, p. 9, 2014.
- [10] C. M. Kamalpriya and P. Singh, "Enhancing program dependency graph based clone detection using approximate subgraph matching," in *2017 IEEE 11th International Workshop on Software Clones (IWSC)*, Klagenfurt, Austria, 2017, pp. 1–7. Available: <http://ieeexplore.ieee.org/document/7880511/>
- [11] J. Kaur, R. Kumar, and S. Kaur, "Design Code Clone Detection System uses Optimal and Intelligence Technique based on Software Engineering," *International Journal of Advanced Research in Computer Science*, p. 7, 2017.
- [12] Y. Sabi, Y. Higo, and S. Kusumoto, "Rearranging the order of program statements for code clone detection," in *2017 IEEE 11th International Workshop on Software Clones (IWSC)*, Klagenfurt, Austria, 2017, pp. 1–7. Available: <http://ieeexplore.ieee.org/document/7880503/>
- [13] H. Kaur and R. Maini, "Performance Evaluation and Comparative Analysis of Code-Clone-Detection Techniques and Tools," *International Journal of Software Engineering and Its Applications*, vol. 11, no. 3, pp. 31–50, Mar. 2017. Available: http://www.sersc.org/journals/IJSEIA/vol11_no3_2017/4.pdf
- [14] L. Li, H. Feng, W. Zhuang, N. Meng, and B. Ryder, "CCLearner: A Deep Learning-Based Clone Detection Approach," in *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Shanghai, 2017, pp. 249–260. Available: <http://ieeexplore.ieee.org/document/8094426/>
- [15] L. Voinea and A. C. Telea, "Visual Clone Analysis with SolidSDD," in *2014 Second IEEE Working Conference on Software Visualization*, Victoria, BC, Canada, 2014, pp. 79–82. Available: <http://ieeexplore.ieee.org/document/6980217/>
- [16] A. Hanjalic, "ClonEvol: Visualizing software evolution with code clones," in *2013 First IEEE Working Conference on Software Visualization (VISSOFT)*, Eindhoven, Netherlands, 2013, pp. 1–4. Available: <http://ieeexplore.ieee.org/document/6650525/>

- [17] Green, P., Lane, P. C. R., Rainer, A. and Scholz, S. B. “Unscrambling Code Clones for One-to-One Matching of Duplicated Code”, Technical Report 502, School of Computer Science, University of Hertfordshire, 2010. [Online]. Available: <https://core.ac.uk/download/pdf/1641752.pdf>
- [18] S. Harris, "Simian: Similarity analyser," 2011. [Online]. Available: <http://www.harukizaemon.com/simian/>.
- [19] K. Hotta, Y. Sasaki, Y. Sano, Y. Higo, and S. Kusumoto, “An Empirical Study on the Impact of Duplicate Code,” *Advances in Software Engineering*, vol. 2012, pp. 1–22, 2012. <https://www.hindawi.com/archive/2012/938296/>
- [20] "The Archive of CCFinder Official Site," 2015. [Online]. Available: <http://www.ccfinder.net/doc/10.2/en/whats.html>
- [21] S. Wagner, A. Abdulkhaleq, K. Kaya, and A. Paar, “On the Relationship of Inconsistent Software Clones and Faults: An Empirical Study,” in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Suita, 2016, pp. 79–89. <http://ieeexplore.ieee.org/document/7476632/>
- [22] Asaduzzaman, M. “Visualization and Analysis of Software Clones”, MSc Thesis, University of Saskatchewan, Saskatoon, 2012. [Online]. Available: https://www.cs.usask.ca/~croy/Theses/Thesis_Asaduzzaman_January2012.pdf
- [23] Koschke, R., Falke, R., Frenzel, P.: Clone detection using abstract syntax suffix trees. In: *Proceedings of the 13th IEEE Working Conference on Reverse Engineering*, Italy, pp. 253–262, October 2006
- [24] Tairas, R., Gray, J.: Phoenix-based clone detection using suffix trees. In: *Proceedings of the 44th ACM Annual Southeast Regional Conference (ACM-SE 2006)*, Melbourne, pp. 679– 684, March 2006
- [25] Wijesiriwardana, C. and Wimalaratne, P., 2017, November. Component-based experimental testbed to facilitate code clone detection research. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 165-168). IEEE.

[26] Wijesiriwardana, C. and Wimalaratne, P., 2018. Fostering Real-Time Software Analysis by Leveraging Heterogeneous and Autonomous Software Repositories. *IEICE TRANSACTIONS on Information and Systems*, 101(11), pp.2730-2743.

[27] Wijesiriwardana, C., Ghezzi, G. and Gall, H., 2012, December. A guided mashup framework for rapid software analysis service composition. In *2012 19th Asia-Pacific Software Engineering Conference (Vol. 1, pp. 725-728)*. IEEE.

[28] Wijesiriwardana, C. and Wimalaratne, P., 2019. Software Engineering Data Analytics: A Framework Based on a Multi-Layered Abstraction Mechanism. *IEICE Transactions on Information and Systems*, 102(3), pp.637-639.