

DAFTAR PUSTAKA

- [1] J. Singh, Abdullah, and M. Bajpai, “Empirical validation of maintainability evaluation model: AI driven approach,” in *Proc. 2025 International Conference on Metaverse and Current Trends in Computing (ICMCTC)*, 2025, doi: 10.1109/ICMCTC62214.2025.11196734.
- [2] T. A. Al-Rawashdeh, F. Al’Azzeh, and F. Al-Tarawneh, “An analytical hierarchy process-based technique for software requirements prioritization,” *IEEE Access*, vol. 13, pp. 50603–50610, 2025, doi: 10.1109/ACCESS.2025.3552490.
- [3] R. Chennappan and K. M. Mounika, “A hybrid machine learning framework for accurate prediction of software maintainability metrics,” in *Proc. 2025 International Conference on Computing Technologies & Data Communication (ICCTDC)*, Hassan, India, Jul. 2025, pp. 1–7, doi: 10.1109/ICCTDC64446.2025.11158018.
- [4] A. Tornhill and M. Borg, “Code Red: The business impact of code quality—A quantitative study of 39 proprietary production codebases,” in *Proc. International Conference on Technical Debt (TechDebt ’22)*, Pittsburgh, PA, USA, May 2022, pp. 11–20, doi: 10.1145/3524843.3528091.
- [5] Y. Tashtoush, N. Abu-El-Rub, O. Darwish, S. Al-Eidi, D. Darweesh, and O. Karajeh, “A notional understanding of the relationship between code readability and software complexity,” *Information*, vol. 14, no. 2, Art. no. 81, Jan. 2023, doi: 10.3390/info14020081.
- [6] Z. Zhang, “The influence of code smell interactions on software maintainability,” in *Proc. 2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2025, pp. 865–869, doi: 10.1109/ICSME64153.2025.00094.
- [7] R. H. Díaz Arrieta, B. L. Díaz Monroy, L. J. Castillo Heredia, and A. G. Valenzuela Cobos, “Global trends and empirical metrics in the evaluation of code smells and technical debt: A bibliometric study,” *IEEE Access*, vol. 13, pp. 138143–138165, 2025, doi: 10.1109/ACCESS.2025.3594657.
- [8] E. Hamouda, A. El-Korany, and S. Makady, “Smell-ML: A machine learning framework for detecting rarely studied code smells,” *IEEE Access*, vol. 13, pp. 12966–12980, 2025, doi: 10.1109/ACCESS.2025.3530927.
- [9] D. Zhang, S. Song, Y. Zhang, H. Liu, and G. Shen, “Code smell detection research based on pre-training and stacking models,” *IEEE Latin America Transactions*, vol. 22, no. 1, pp. 22–30, Jan. 2024.
- [10] D. Nikolić, D. Stefanović, M. Nikolić, D. Dakić, M. Stefanović, and S. Koprivica, “Uncovering determinants of code quality in education via static code analysis,” *IEEE Access*, vol. 12, pp. 168229–168244, 2024, doi: 10.1109/ACCESS.2024.3426299.
- [11] T. Sharma, “Multi-faceted code smell detection at scale using DesigniteJava 2.0,” in *Proc. 21st International Conference on Mining Software Repositories (MSR ’24)*, Lisbon, Portugal, Apr. 2024, pp. 284–288, doi: 10.1145/3643991.3644881.
- [12] D. J. Thessalonica, H. K. Nehemiah, S. Sreejith, and A. Kannan, “Intelligent mining of association rules based on nanopatterns for code smells detection,” *Scientific*

- Programming*, vol. 2023, Art. no. 2973250, pp. 1–18, 2023, doi: 10.1155/2023/2973250.
- [13] R. D. Novendra and W. D. Sunindyo, “Emerging trends in code quality: Introducing Kotlin-specific bad smell detection tool for Android apps,” *IEEE Access*, vol. 12, pp. 63895–63903, 2024, doi: 10.1109/ACCESS.2024.3397055.
- [14] A. Gupta and N. K. Chauhan, “A severity-based classification assessment of code smells in Kotlin and Java application,” *Arabian Journal for Science and Engineering*, vol. 47, pp. 1831–1848, 2022, doi: 10.1007/s13369-021-06077-6.
- [15] I. Kaur and A. Kaur, “A novel four-way approach designed with ensemble feature selection for code smell detection,” *IEEE Access*, vol. 9, pp. 8695–8707, 2021, doi: 10.1109/ACCESS.2021.3049823.
- [16] T. Thaher, M. Mafarja, H. Turabieh, P. A. Castillo, H. Faris, and I. Aljarah, “Teaching learning-based optimization with evolutionary binarization schemes for tackling feature selection problems,” *IEEE Access*, vol. 9, pp. 41082–41103, 2021, doi: 10.1109/ACCESS.2021.3064799.
- [17] F. Pecorelli, S. Lujan, V. Lenarduzzi, F. Palomba, and A. De Lucia, “On the adequacy of static analysis warnings with respect to code smell prediction,” *Empirical Software Engineering*, vol. 27, Art. no. 64, 2022, doi: 10.1007/s10664-022-10126-5.
- [18] M. Y. Mhawish and M. Gupta, “Predicting code smells and analysis of predictions: Using machine learning techniques and software metrics,” *Journal of Computer Science and Technology*, vol. 35, no. 6, pp. 1428–1445, Nov. 2020, doi: 10.1007/s11390-020-0323-7.
- [19] R. Coppola, T. Fulcini, L. Ardito, and M. Torchiano, “Kotlin assimilating the Android ecosystem: An appraisal of diffusion and impact on maintainability,” *Journal of Systems and Software*, vol. 222, Art. no. 112346, 2025, doi: 10.1016/j.jss.2025.112346.
- [20] A. Santana, E. Figueiredo, and J. A. Pereira, “Unraveling the impact of code smell agglomerations on code stability,” in *Proc. 2024 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2024, pp. 461–473, doi: 10.1109/ICSME58944.2024.00049.
- [21] O. A. Olusola, I. O. Adewumi, and B. R. Ariyo, “Overview of bad code smells in software development and researches,” *Journal of Robotics and Automation Research*, vol. 6, no. 4, pp. 1–11, Nov. 2025.
- [22] L. Newcomb, A. Newcomb, and O. Ochoa, “Preconditions and postconditions as design constraints for LLM code generation,” *IEEE Access*, vol. 13, pp. 184259–184274, 2025, doi: 10.1109/ACCESS.2025.3625819.
- [23] M. G. Al-Obeidallah, “Combining structural features and software metrics for automated design pattern detection,” *IEEE Access*, vol. 13, pp. 177671–177680, 2025, doi: 10.1109/ACCESS.2025.3620144.
- [24] M. A. A. Hilmi, A. Puspiningrum, Darsih, D. O. Siahaan, H. S. Samosir, and A. S. Rahma, “Research trends, detection methods, practices, and challenges in code smell: SLR,” *IEEE Access*, vol. 11, pp. 129536–129551, 2023, doi: 10.1109/ACCESS.2023.3334258.

- [25] D. Albuquerque, E. Guimarães, M. Perkusich, T. Rique, F. Cunha, H. Almeida, and A. Perkusich, "On the assessment of interactive detection of code smells in practice: A controlled experiment," *IEEE Access*, vol. 11, pp. 84589–84606, 2023, doi: 10.1109/ACCESS.2023.3302260.
- [26] P. S. Yadav, R. S. Rao, A. Mishra, and M. Gupta, "Machine learning-based methods for code smell detection: A survey," *Applied Sciences*, vol. 14, no. 14, Art. no. 6149, Jul. 2024, doi: 10.3390/app14146149.
- [27] Khyber, S. Ali, F. Wahid, S. Baseer, A. Alkhayyat, and A. M. Al-Radaei, "Smell-aware bug classification," *IEEE Access*, vol. 12, pp. 14061–14082, 2024, doi: 10.1109/ACCESS.2023.3335175.
- [28] H. B. Hassan, Q. I. Sarhan, and Á. Beszédes, "Evaluating Python static code analysis tools using FAIR principles," *IEEE Access*, vol. 12, pp. 173647–173659, 2024, doi: 10.1109/ACCESS.2024.3503493.
- [29] S. D. Garomssa, R. Kannan, I. Chai, and D. Riehle, "How software quality mediates the impact of intellectual capital on commercial open-source software company success," *IEEE Access*, vol. 10, pp. 46490–46503, 2022, doi: 10.1109/ACCESS.2022.3170058.
- [30] Gintoro, F. L. Gaol, A. N. Fajar, A. S. Girsang, and T. Matsuo, "Microservices design pattern in action: Improving modifiability in microservices-based software development," *IEEE Access*, vol. 13, pp. 163588–163606, 2025, doi: 10.1109/ACCESS.2025.3610272.
- [31] D. Gobov and O. Zuieva, "Software quality attributes in requirements engineering," *International Journal of Information Technology and Computer Science*, vol. 17, no. 4, pp. 38–48, Aug. 2025, doi: 10.5815/ijitcs.2025.04.04.
- [32] M. H. Alshahrani, M. S. Maashi, and A. B. Najjar, "Architectural styles and quality attributes in AI-based video software: A systematic literature review," *IEEE Access*, vol. 13, pp. 185296–185317, 2025, doi: 10.1109/ACCESS.2025.3625266.
- [33] I. L. Castellano, G. F. C. Aguilar, N. Silega, T. Kamal, M. S. A. M. Al-Gaashani, N. Abdel Samee, and M. Alabdulhafith, "An ontology-based approach to reduce the negative impact of code smells in software development projects," *IEEE Access*, vol. 11, pp. 100146–100153, 2023, doi: 10.1109/ACCESS.2023.3300575.
- [34] T. Heričko and B. Šumak, "Exploring maintainability index variants for software maintainability measurement in object-oriented systems," *Applied Sciences*, vol. 13, no. 5, Art. no. 2972, Feb. 2023, doi: 10.3390/app13052972.
- [35] M. Klima, M. Bures, K. Frajtek, V. Rechtberger, M. Trnka, X. Bellekens, T. Cerny, and B. S. Ahmed, "Selected code-quality characteristics and metrics for Internet of Things systems," *IEEE Access*, vol. 10, pp. 46144–46161, 2022, doi: 10.1109/ACCESS.2022.3170475.
- [36] R. A. Halepmollasi and A. Tosun, "Predicting vulnerable code changes using technical debt indicators," *IEEE Access*, vol. 13, pp. 153896–153917, 2025, doi: 10.1109/ACCESS.2025.3602480.
- [37] N. Nikolaidis, D. Zisis, A. Ampatzoglou, A. Chatzigeorgiou, and D. Soudris, "Experience with managing technical debt in scientific software development using

- the EXA2PRO framework,” *IEEE Access*, vol. 9, pp. 72524–72534, 2021, doi: 10.1109/ACCESS.2021.3079271.
- [38] D. Cui, “Early detection of flawed structural dependencies during software evolution,” *IEEE Access*, vol. 9, pp. 28856–28871, 2021, doi: 10.1109/ACCESS.2021.3054472.
- [39] E. A. AlOmar, A. Peruma, M. W. Mkaouer, C. Newman, A. Ouni, and M. Kessentini, “How we refactor and how we document it? On the use of supervised machine learning algorithms to classify refactoring documentation,” *Expert Systems with Applications*, vol. 167, Art. no. 114176, 2021, doi: 10.1016/j.eswa.2020.114176.
- [40] T. Alkhaeir and B. Walter, “The effect of code smells on the relationship between design patterns and defects,” *IEEE Access*, vol. 9, pp. 3360–3373, 2021, doi: 10.1109/ACCESS.2020.3047870.
- [41] M. A. Alkandari, A. Kelkawi, and M. O. Elish, “An empirical investigation on the effect of code smells on resource usage of Android mobile applications,” *IEEE Access*, vol. 9, pp. 61853–61863, 2021, doi: 10.1109/ACCESS.2021.3075040.
- [42] P. Flores, M. Alvarez, and J. Torres, “Identifying difficulties of software modeling through class diagrams: A long-term comparative analysis,” *IEEE Access*, vol. 10, pp. 28895–28910, 2022, doi: 10.1109/ACCESS.2022.3157290.
- [43] M. Fawad, G. Rasool, and M. B. Riaz, “Refactoring Android source code smells from Android applications,” *IEEE Access*, vol. 13, pp. 14122–14150, 2025, doi: 10.1109/ACCESS.2025.3529687.
- [44] F. Yang, H. Xu, P. Xiao, F. Zhong, and G. Zeng, “A method-level defect prediction approach based on structural features of method-calling network,” *IEEE Access*, vol. 11, pp. 7933–7946, 2023, doi: 10.1109/ACCESS.2023.3239266.
- [45] T. Akinyemi, E. Solomon, A. Woubie, and K. Lippert, “A comprehensive review of static memory analysis,” *IEEE Access*, vol. 12, pp. 170204–170226, 2024, doi: 10.1109/ACCESS.2024.3482253.
- [46] Y. Bugayenko, K. Daniakin, M. Farina, Z. Kholmatova, A. Kruglov, W. Pedrycz, and G. Succi, “Qualitative clustering of software repositories based on software metrics,” *IEEE Access*, vol. 11, pp. 14716–14727, 2023, doi: 10.1109/ACCESS.2023.3244495.
- [47] S. Alawadi, K. Alkharabsheh, F. Alkhabbas, V. R. Kebande, F. M. Awaysheh, F. Palomba, and M. Awad, “FedCSD: A federated learning based approach for code-smell detection,” *IEEE Access*, vol. 12, pp. 44888–44904, 2024, doi: 10.1109/ACCESS.2024.3380167.
- [48] S. S. Ahamad, “A novel NFC-based secure protocol for merchant transactions,” *IEEE Access*, vol. 10, pp. 1905–1920, 2022, doi: 10.1109/ACCESS.2021.3139065.
- [49] N. Ito, M. Hashimoto, and A. Otsuka, “Feature extraction methods for binary code similarity detection using neural machine translation models,” *IEEE Access*, vol. 11, pp. 102796–102805, 2023, doi: 10.1109/ACCESS.2023.3316215.
- [50] X. He, A. H. Lashkari, N. Vombatkere, and D. P. Sharma, “Authorship attribution methods, challenges, and future research directions: A comprehensive survey,” *Information*, vol. 15, no. 3, Art. no. 131, 2024, doi: 10.3390/info15030131.

- [51] H.-J. Choi and H.-S. Chae, "Development of testability prediction models considering complexity diversity for C programs," *IEEE Access*, vol. 11, pp. 98469–98485, 2023, doi: 10.1109/ACCESS.2023.3312556.
- [52] L. Al-Shalabi, "New feature selection algorithm based on feature stability and correlation," *IEEE Access*, vol. 10, pp. 4699–4713, 2022, doi: 10.1109/ACCESS.2022.3140209.
- [53] D. A. Elmanakhly, M. Saleh, E. A. Rashed, and M. Abdel-Basset, "BinHOA: Efficient binary horse herd optimization method for feature selection: Analysis and validations," *IEEE Access*, vol. 10, pp. 26795–26816, 2022, doi: 10.1109/ACCESS.2022.3156593.
- [54] W. Zhang, Y. Zhao, C. Yuan, X. Chen, W. Liu, Y. Feng, Y. Feng, J. Yang, and M. Li, "Dynamic multi-level competition learning-based dual-task optimization for high-dimensional feature selection," *IEEE Access*, vol. 12, pp. 190347–190361, 2024, doi: 10.1109/ACCESS.2024.3510888.
- [55] M. R. Islam, A. A. Lima, S. C. Das, M. F. Mridha, A. R. Prodeep, and Y. Watanobe, "A comprehensive survey on the process, methods, evaluation, and challenges of feature selection," *IEEE Access*, vol. 10, pp. 99595–99632, 2022, doi: 10.1109/ACCESS.2022.3205618.
- [56] H. C. S. C. Lima, F. E. B. Otero, L. H. C. Merschmann, and M. J. F. Souza, "A novel hybrid feature selection algorithm for hierarchical classification," *IEEE Access*, vol. 9, pp. 127278–127292, 2021, doi: 10.1109/ACCESS.2021.3112396.
- [57] L. Zhang and X. Chen, "Feature selection methods based on symmetric uncertainty coefficients and independent classification information," *IEEE Access*, vol. 9, pp. 13845–13858, 2021, doi: 10.1109/ACCESS.2021.3049815.
- [58] E. Sreehari and L. D. Dhinesh Babu, "Feature selection techniques for data analysis and decision making in interdisciplinary areas: A systematic review," *IEEE Access*, vol. 12, pp. 188845–188873, 2024, doi: 10.1109/ACCESS.2024.3512878.
- [59] O. M. Alyasiri, Y.-N. Cheah, A. K. Abasi, and O. M. Al-Janabi, "Wrapper and hybrid feature selection methods using metaheuristic algorithms for English text classification: A systematic review," *IEEE Access*, vol. 10, pp. 39833–39852, 2022, doi: 10.1109/ACCESS.2022.3165814.
- [60] J.-G. Choi, I. Ko, and S. Han, "Depression level classification using machine learning classifiers based on actigraphy data," *IEEE Access*, vol. 9, pp. 116622–116646, 2021, doi: 10.1109/ACCESS.2021.3105393.
- [61] A. A. Linkon, I. R. Noman, M. R. Islam, J. C. Bortty, K. K. Bishnu, A. Islam, R. Hasan, and M. Abdullah, "Evaluation of feature transformation and machine learning models on early detection of diabetes mellitus," *IEEE Access*, vol. 12, pp. 165425–165440, 2024, doi: 10.1109/ACCESS.2024.3488743.
- [62] E. H. Sayed, A. Alabrah, K. H. Rahouma, M. Zohaib, and R. M. Badry, "Machine learning and deep learning for loan prediction in banking: Exploring ensemble methods and data balancing," *IEEE Access*, vol. 12, pp. 193997–194019, 2024, doi: 10.1109/ACCESS.2024.3509774.

- [63] M. Altalhan, A. Algarni, and M. Turki-Hadj Alouane, “Imbalanced data problem in machine learning: A review,” *IEEE Access*, vol. 13, pp. 13686–13699, 2025, doi: 10.1109/ACCESS.2025.3531662.
- [64] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, “Comparative analysis using various performance metrics in imbalanced data for multi-class text classification,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, pp. 1082–1090, 2023, doi: 10.14569/IJACSA.2023.01406116.
- [65] P. Fränti and R. Marescu-Istodor, “Soft precision and recall,” *Pattern Recognition Letters*, vol. 167, pp. 115–121, 2023, doi: 10.1016/j.patrec.2023.02.005.
- [66] K. Takahashi, K. Yamamoto, A. Kuchiba, A. Shintani, and T. Koyama, “Hypothesis testing procedure for binary and multi-class F1-scores in the paired design,” *Statistics in Medicine*, vol. 42, no. 23, pp. 4177–4192, Oct. 2023, doi: 10.1002/sim.9853.
- [67] S. Simhadri, M. Ponnamm, R. Balamurugan, and R. Rajitha, “Enhanced multi-class model evaluation: Analyzing BERT, GPT-2, and LLaMA with precision, recall, and F1-score metrics,” in *Proc. 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2025, pp. 984–989, doi: 10.1109/ICIMIA67127.2025.11200914.
- [68] F. Palomba, G. Bavota, M. Di Penta, F. Fasano, R. Oliveto, and A. De Lucia, “A large-scale empirical study on the lifecycle of code smell co-occurrences,” *Information and Software Technology*, vol. 99, pp. 1–10, 2018, doi: 10.1016/j.infsof.2018.02.004.
- [69] R. Krishnamurthy, G. Piskachev, and E. Bodden, “To What Extent Can We Analyze Kotlin Programs Using Existing Java Taint Analysis Tools?,” in *Proc. IEEE 22nd International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2022, pp. 230–235, doi: 10.1109/SCAM55253.2022.00032.
- [70] A. Nguyen-Duc, M. V. Do, Q. L. Hong, K. N. Khac, and A. N. Quang, “On the adoption of static analysis for software security assessment—A case study of an open-source e-government project,” *Computers & Security*, vol. 111, Art. no. 102470, 2021, doi: 10.1016/j.cose.2021.102470.
- [71] C. Liang, Q. Wei, J. Du, Y. Wang, and Z. Jiang, “Survey of source code vulnerability analysis based on deep learning,” *Computers & Security*, vol. 148, Art. no. 104098, 2025, doi: 10.1016/j.cose.2024.104098.
- [72] P. Nunes, J. Fonseca, and M. Vieira, “Blending static and dynamic analysis for web application vulnerability detection: Methodology and case study,” *IEEE Access*, vol. 13, pp. 3139–3155, 2025, doi: 10.1109/ACCESS.2024.3522094.
- [73] F. Pecorelli, S. Lujan, V. Lenarduzzi, F. Palomba, and A. De Lucia, “On the adequacy of static analysis warnings with respect to code smell prediction,” *Empirical Software Engineering*, vol. 27, Art. no. 64, 2022, doi: 10.1007/s10664-022-10126-5.
- [74] T. Mahmud, M. Che, A. Ngu, and G. Yang, “Why Android app testing falls short: Empirical insights from open-source projects and a practitioner survey,” *Empirical Software Engineering*, vol. 30, Art. no. 163, 2025, doi: 10.1007/s10664-025-10726-x.

- [75] M. Auer, I. A. Moreno, and G. Fraser, “WallMauer: Robust code coverage instrumentation for Android apps,” in *Proc. 5th ACM/IEEE International Conference on Automation of Software Test (AST '24)*, Lisbon, Portugal, 2024, pp. 34–44, doi: 10.1145/3644032.3644462.
- [76] A. Wibowo, A. E. Permanasari, and T. B. Adji, “Enhanced code smell detection using random forest optimized with particle swarm variants,” *Results in Engineering*, vol. 29, Art. no. 109729, 2026, doi: 10.1016/j.rineng.2026.109729.
- [77] F-Droid, “F-Droid - Free and Open Source Android App Repository.” [Online]. Available: <https://f-droid.org/en/>
- [78] F-Droid, “Apps.” [Online]. Available: <https://f-droid.org/en/packages/>
- [79] Detekt, “Complexity Rule Set: LongMethod.” [Online]. Available: <https://detekt.dev/docs/rules/complexity/>