**REFERENCES**

[1] A. O. Christiana, B. A. Gyunka, and A. Noah, “Android Malware Detection through Machine Learning Techniques: A Review,” Int. J. Online Biomed. Eng. IJOE, vol. 16, no. 02, p. 14, Feb. 2020, doi: 10.3991/ijoe.v16i02.11549. [2] D. Ghimire and J. Lee, “Geometric Feature-Based Facial Expression Recognition in Image Sequences Using Multi-Class AdaBoost and Support Vector Machines,” Sensors, vol. 13, no. 6, pp. 7714–7734, Jun. 2013, doi: 10.3390/s130607714. [3] R. Wang, “AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review,” Phys. Procedia, vol. 25, pp. 800–807, 2012, doi: 10.1016/j.phpro.2012.03.160. [4] J. Sun, H. Fujita, P. Chen, and H. Li, “Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble,” Knowl.-Based Syst., vol. 120, pp. 4–14, Mar. 2017, doi: 10.1016/j.knosys.2016.12.019. [5] A. Garg and K. Tai, “Comparison of statistical and machine learning methods in modelling of data with multicollinearity,” Int. J. Model. Identif. Control, vol. 18, no. 4, p. 295, 2013, doi: 10.1504/IJMIC.2013.053535. [6] C. P. Obite, N. P. Olewuezi, G. U. Ugwuanyim, and D. C. Bartholomew, “Multicollinearity Effect in Regression Analysis: A Feed Forward Artificial Neural Network Approach,” Asian J. Probab. Stat., pp. 22–33, Jan. 2020, doi: 10.9734/ajpas/2020/v6i130151. [7] W. Wang et al., “Constructing Features for Detecting Android Malicious Applications: Issues, Taxonomy and Directions,” IEEE Access, vol. 7, pp. 67602–67631, 2019, doi: 10.1109/ACCESS.2019.2918139. [8] B. Rashidi, C. Fung, and E. Bertino, “Android malicious application detection using support vector machine and active learning,” in 2017 13th International Conference on Network and Service Management (CNSM), Tokyo, Nov. 2017, pp. 1–9. doi: 10.23919/CNSM.2017.8256035. [9] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, “Significant Permission Identification for Machine-Learning-Based Android Malware Detection,” IEEE Trans. Ind. Inform., vol. 14, no. 7, pp. 3216–3225, Jul. 2018, doi: 10.1109/TII.2017.2789219. [10] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco, “Dendroid: A text mining approach to analyzing and classifying code structures in Android malware families,” Expert Syst. Appl., vol. 41, no. 4, pp. 1104–1117, Mar. 2014, doi: 10.1016/j.eswa.2013.07.106. [11] M. Magdum, “Permission based Mobile Malware Detection System using Machine Learning Techniques,” vol. 14, no. 6, pp. 6170–6174, 2015. [12] M. Qiao, A. H. Sung, and Q. Liu, “Merging Permission and API Features for Android Malware Detection,” in 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Kumamoto, Japan, Jul. 2016, pp. 566–571. doi: 10.1109/IIAI- AAI.2016.237. [13] D. O. Sahin, O. E. Kural, S. Akleylek, and E. Kilic, “New results on permission based static analysis for Android malware,” in 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, Mar. 2018, pp. 1–4. doi: 10.1109/ISDFS.2018.8355377. [14] A. Mahindru and A. L. Sangal, “MLDroid—framework for Android malware detection using machine learning techniques,” Neural Comput. Appl., vol. 33, no. 10, pp. 5183–5240, May 2021, doi: 10.1007/s00521-020-05309-4. [15] X. Su, D. Zhang, W. Li, and K. Zhao, “A Deep Learning Approach to Android Malware Feature Learning and Detection,” in 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, Aug. 2016, pp. 244–251. doi: 10.1109/TrustCom.2016.0070. [16] K. A. Talha, D. I. Alper, and C. Aydin, “APK Auditor: Permission-based Android malware detection system,” Digit. Investig., vol. 13, pp. 1–14, Jun. 2015, doi: 10.1016/j.diin.2015.01.001. [17] A. Mahindru and P. Singh, “Dynamic Permissions based Android Malware Detection using Machine Learning Techniques,” in Proceedings of the 10th Innovations in Software Engineering Conference, Jaipur India, Feb. 2017, pp. 202–210. doi: 10.1145/3021460.3021485. [18] U. Pehlivan, N. Baltaci, C. Acarturk, and N. Baykal, “The analysis of feature selection methods and classification algorithms in permission based Android malware detection,” in 2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Orlando, FL, USA, Dec. 2014, pp. 1–8. doi: 10.1109/CICYBS.2014.7013371. [19] M. Kedziora, P. Gawin, M. Szczepanik, and I. Jozwiak, “Malware Detection Using Machine Learning Algorithms and Reverse Engineering of Android Java Code,” Int. J. Netw. Secur. Its Appl., vol. 11, no. 01, pp. 01–14, Jan. 2019, doi: 10.5121/ijnsa.2019.11101. [20] X. Liu and J. Liu, “A Two-Layered Permission-Based Android Malware Detection Scheme,” in 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Oxford, United Kingdom, Apr. 2014, pp. 142–148. doi: 10.1109/MobileCloud.2014.22. [21] “Permission-Based Android Malware Detection | Semantic Scholar.” https://www.semanticscholar.org/paper/Permission-Based-Android- Malware-Detection-Aung- Zaw/c8576b5df33813fe8938cbb19e35217ee21fc80b (accessed Oct. 31, 2021). [22] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, “Drebin: Effective and Explainable Detection of Android Malware in Your Pocket,” presented at the Network and Distributed System Security Symposium, San Diego, CA, 2014. doi: 10.14722/ndss.2014.23247. [23] H. Cai, N. Meng, B. Ryder, and D. Yao, “DroidCat: Effective Android Malware Detection and Categorization via App-Level Profiling,” IEEE Trans. Inf. Forensics Secur., vol. 14, no. 6, pp. 1455–1470, Jun. 2019, doi: 10.1109/TIFS.2018.2879302. [24] P. Rovelli and Ý. Vigfússon, “PMDS: Permission-Based Malware Detection System,” in Information Systems Security, vol. 8880, A. Prakash and R. Shyamasundar, Eds. Cham: Springer International Publishing, 2014, pp. 338–357. doi: 10.1007/978-3-319-13841-1\_19. [25] M. S. Alam and S. T. Vuong, “Random Forest Classification for Detecting Android Malware,” in 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, Aug. 2013, pp. 663–669. doi: 10.1109/GreenCom-iThings- CPSCom.2013.122.