Detecção de Malwares Android: *datasets* e reprodutibilidade

**Taina Soares[1], Guilherme Siqueira[1], Lucas Barcellos[1], Renato Sayyed[1]
Luciano Vargas[1], Gustavo Rodrigues[1], Joner Assolin[1], Jonas Pontes[2],
Eduardo Feitosa[2], Diego Kreutz[1]**

[1] Universidade Federal do Pampa (Unipampa)
[2]Universidade Federal do Amazonas (UFAM)

{{NomeSobrenome}.aluno,diegokreutz}@unipampa.edu.br
{pontes,efeitosa}@icomp.ufam.edu.br

Resumo. *Neste trabalho nós avaliamos uma amostra inicial de 38 trabalhos de pesquisa que utilizam aprendizado de máquina para detecção de malwares Android. Analisamos, em particular, o detalhamento e a disponibilidade dos datasets, que são cruciais para a validação e a reprodutibilidade do trabalho. Nossos resultados sugerem que 100% das pesquisas não são reprodutíveis por falta de informações e/ou acesso aos dados originais da pesquisa.*

1. Introdução

Os modelos de aprendizado de máquina para classificar os aplicativos Android, empacotados como APKs, entre malignos e benignos são os mais utilizados na literatura e na prática [Arslan et al., 2019]. Um modelo preditivo classifica os aplicativos de acordo com premissas que aprendeu durante a fase de treinamento, que ocorre através das características dos aplicativos organizadas como um conjunto estruturado de dados, conhecido como *dataset*. Consequentemente, a apresentação detalhada e a disponibilidade do *dataset* é imprescindível para a validação e a reprodução de trabalhos de detecção de *malwares* [Kouliaridis et al., 2020].

Neste trabalho, o objetivo é avaliarmos a reprodutibilidade, com base nos *datasets* utilizados, de estudos que propõem métodos de aprendizado de máquina para a detecção de *malwares* Android. Para alcançá-lo, coletamos 38 trabalhos existentes na literatura e realizamos um levantamento sobre a disponibilidade e o nível de detalhamento dos *datasets*.

Como contribuições deste trabalho podemos destacar: (i) realização de um levantamento inicial sobre o detalhamento dos *datasets*; (ii) mapeamento detalhado da disponibilidade dos *datasets*; (iii) identificação de incompletude e inconsistências nos trabalhos; (iv) recomendações de boas práticas para trabalhos de pesquisa que utilizem métodos de aprendizado de máquina.

O restante do trabalho está organizado da seguinte forma. Nas Seções 2 e 3 apresentamos e discutimos o levantamento de dados dos 38 trabalhos analisados. Na Seção 4 apresentamos recomendações e as considerações finais. É importante também destacar que apresentamos dados e detalhamentos adicionais na versão estendida do trabalho [Soares et al., 2021], incluindo observações empíricas gerais e o detalhamento das amostras dos *datasets* de cada um dos 38 trabalhos analisados.

2. Metodologia

Para realizar este estudo, selecionamos artigos de diferentes fontes, classificados em quatro grupos: *Grupo 1 (G1)* contém os trabalhos citados por algum *survey* ou revisão sistemática de literatura específica do tema; *Grupo 2 (G2)* inclui trabalhos com 40 (ou mais) citações segundo o Google Scholar (`https://scholar.google.com`); *Grupo 3 (G3)* contém aqueles publicados nos principais periódicos ou conferências da área de segurança, segundo o `Guide2Research.com`; e *Grupo 4 (G4)* inclui artigos publicados em conferências específicas da área de inteligência artificial. Com este último grupo, o objetivo é verificar se existe alguma diferença qualitativa significativa em termos de descrição e disponibilidade das fontes dos *datasets* quando o trabalho é publicado nessa área específica da computação, que engloba o aprendizado de máquina.

Dos 38 trabalhos que compõem este estudo, 6 são artigos retirados de revisões sistemáticas [Sharma and Rattan, 2021, Kumars et al., 2021] (*G1*). Para o grupo *G2*, resultado de uma busca no Google Scholar por "malware detection Android machine learning", foram selecionados os 14 primeiros resultados com 40 (ou mais) citações. Por fim, para os grupos *G3* e *G4*, foram selecionados 12 trabalhos publicados nas principais conferências e periódicos da área de segurança e 6 trabalhos publicados em conferências e periódicos de inteligência artificial, respectivamente.

A análise dos 38 trabalhos ocorreu em duas etapas. Na primeira, cada artigo foi analisado por dois ou três co-autores (revisores). Na segunda etapa, os artigos que resultaram em análises divergentes na primeira etapa foram novamente verificados, desta vez por um, dois ou três revisores diferentes de acordo com a complexidade das divergências. A análise de cada artigo foi guiada pelas seguintes questões: (a) Qual(is) a(s) fonte(s) de dados utilizada(s) na construção do *dataset*?; (b) A fonte de dados, que serviu como origem para os dados, é acessível? Se sim, de qual forma?; (c) Quais informações específicas (*e.g.*, quantidade, nomes, versões) sobre as aplicações Android que compõem o *dataset* são mencionadas no trabalho?

3. Resultados e Discussão

A Tabela 1 resume as informações de origem e disponibilidade dos dados dos *datasets* dos trabalhos analisados. A *Informação da origem* simplesmente registra a menção da origem dos dados nos trabalhos analisados, isto é, se o trabalho informou as fontes das quais retirou todos os dados que utilizou, definimos a coluna como *Sim*. Se apenas parte das fontes dos dados (*e.g.* de aplicações maliciosas ou benignas) foi informada, definimos como *Parcial*. E se o trabalho não informou qualquer origem dos dados, definimos como *Não*.

3.1. Detalhamento dos *datasets*

Durante a análise dos trabalhos, um dos objetivos foi identificar o nível de detalhamento da descrição dos *datasets* utilizados, mais especificamente a existência ou a ausência de informações como: (a) referência à origem das amostras utilizadas, sejam estas oriundas de um *dataset* existente ou extraídas de APKs disponíveis em um repositório; (b) detalhamento da quantidade de amostras utilizadas em cada experimento realizado; e (c) descrição da forma como o conjunto de dados próprio do trabalho foi criado (*e.g.*,

Tabela 1. Detalhamento da origem e disponibilidade dos *datasets*

Papers	Grupo	Informação da origem	Dados disponíveis
[Zhu et al., 2018], [Ali et al., 2017],	G1	Sim	Sim
[Alazab et al., 2020]	G2		
[Pendlebury et al., 2019]	G3		
[Vinod et al., 2019], [Kabakus and Dogru, 2018]	G1	Sim	Parcial
[Yuan et al., 2016], [Mahindru and Singh, 2017], [Amos et al., 2013], [Yuan et al., 2014]	G2		
[Demontis et al., 2019], [Cen et al., 2015], [Gates et al., 2014], [Ferrante et al., 2018]	G3		
[Jung et al., 2018]	G4		
[Patel and Buddadev, 2015]	G1	Parcial	Parcial
[Arora et al., 2018]	G1	Sim	Não
[Ma et al., 2019] , [Yerima et al., 2014] , [Li et al., 2018], [Mas'ud et al., 2014] , [Narudin et al., 2016]	G2		
[Chawla et al., 2021], [Fan et al., 2017], [Chen et al., 2020], [Jordaney et al., 2017], [Li et al., 2021], [Xu et al., 2016]	G3		
[Arslan et al., 2019], [Peiravian and Zhu, 2013] [Chen et al., 2018], [Mahindru and Sangal, 2021]	G4		
[Wang et al., 2019]	G1	Parcial	Não
[Wu and Hung, 2014],	G2		
[Burguera et al., 2011]	G3		
[Shabtai et al., 2012]	G4		
[Sahs and Khan, 2012], [Zarni Aung, 2013]	G2	Não	Não

combinação de *subsets* de outros *datasets*), aplicável quando um estudo utiliza particionamentos não detalhados de outros conjuntos de dados ou desenvolve suas próprias amostras.

O item (c) representa o nível mais completo de detalhamento dos *datasets*. Para que um trabalho satisfaça esse item, ele deve fornecer, além da origem dos dados e as quantidades de amostras - itens (a) e (b), um detalhamento específico dessas amostras, como os nomes e as versões das aplicações. Apesar de existirem repositórios de APKs voltados para o desenvolvimento de métodos de detecção de *malwares*, como o AndroZoo (`https://androzoo.uni.lu`), no qual são disponibilizados os nomes dos aplicativos e os resumos criptográficos, nenhum dos trabalhos analisados - nem aqueles que utilizam *subsets* de outros *datasets*, nem aqueles que desenvolvem as próprias amostras fornece essas informações necessárias para a sua reprodutibilidade.

Observando a Tabela 1, podemos visualizar as deficiências no detalhamento dos *datasets* quanto ao item (a). Embora dados referentes aos itens (b) e (c) não estejam na tabela[1], ao levarmos em consideração os itens (a) e (b), bem como a disponibilidade das fontes de dados utilizadas, aproximadamente 90% dos estudos não detalham suficientemente a origem do conjunto de dados utilizado ou não utilizam fontes disponíveis. Do total de trabalhos analisados, apenas 4 (apontados nas três primeiras linhas da tabela) mencionam a origem dos dados, utilizam fontes disponíveis e informam a quantidade de

[1]A inclusão dos itens (b) e (c) na tabela inviabilizaria o agrupamento dos trabalhos. Ao consideramos também a limitação de espaço, optamos por não representar estes itens na tabela.

amostras benignas e de *malwares* que compõem os *datasets*.

Em 12 trabalhos (aproximadamente 32%), a informação faltante é referente à quantidade de aplicativos (item b), utilizados no *dataset*, que são oriundos de lojas de aplicativos (*e.g.*, Google Play Store, AppChina, Mumayi, Amazon Appstore) ou *datasets* (*e.g.*, The Drebin Dataset, DroidKin, ContagioDump). A informação referente ao item (b) pode ser vista na tabela detalhada da versão estendida do trabalho [Soares et al., 2021]. Por exemplo, há trabalhos, como [Alazab et al., 2020], que informam a origem dos dados, mas não identificam a quantidade e nem o nome (ou resumo criptográfico) dos aplicativos retirados de cada fonte de dados. Além disso, trabalhos como [Sahs and Khan, 2012, Zarni Aung, 2013] informam o número de amostras e a distribuição do total delas em cada classe (*i.e.*, maligno ou benigno), mas não especificam a origem dos dados.

3.2. Origem dos dados

Em 60% dos trabalhos, a origem dos aplicativos benignos são lojas online de aplicativos (*e.g.*, Google Play Store, Chinese Market, Amazon Appstore App For Android, APKPure App)[2]. Entretanto, para a reconstrução do *dataset*, seriam necessárias informações como o nome e a versão dos aplicativos retirados dessas lojas. Infelizmente, nenhum dos trabalhos fornece esses detalhes.

3.3. Disponibilidade da fonte dos dados

Dos trabalhos analisados e que mencionam pelo menos alguma origem de dados, apenas quatro ([Pendlebury et al., 2019], [Ali et al., 2017], [Zhu et al., 2018], [Alazab et al., 2020]) possuem todas as origens disponíveis. As fontes de dados citadas por estes são AndroZoo, ContagioDump, MalShare, VirusShare e M0Droid[3].

Em aproximadamente 58% dos trabalhos, aqueles em que, na Tabela 1, a coluna *Dados disponíveis* está como *Não*, as fontes referenciadas são inacessíveis, como é o caso de trabalhos como [Jordaney et al., 2017] e [Chawla et al., 2021]. É interessante destacarmos também que alguns trabalhos, como [Shabtai et al., 2012], relatam que as amostras utilizadas no experimento foram desenvolvidas internamente, porém sem fornecer detalhes ou o acesso à tais amostras. Em todos esses casos, temos problemas que afetam a reprodutibilidade dos trabalhos, como é evidente.

4. Considerações Finais

A partir da análise minuciosa de 38 *papers*, podemos concluir que todos os trabalhos falham em apresentar pelo menos alguma informação fundamental acerca dos *datasets* (*e.g.*, origem dos dados, quantidade de aplicativos) ou não indicam a forma de acessar a fonte de dados utilizada na construção do *dataset*. Resumidamente, podemos assumir que os dados coletados indicam que a maioria das pesquisas em detecção de *malwares* Android não são reprodutíveis e nem verificáveis devido à falta de informação sobre os dados utilizados. Esse cenário traz impactos negativos, por exemplo, na construção de

[2]`https://play.google.com/store`, `https://shouji.baidu.com/`, `https://www.amazon.com/gp/mas/get/amazonapp`, `https://m.apkpure.com`

[3]`http://contagiominidump.blogspot.com/`, `https://malshare.com/`, `https://virusshare.com/`, `https://www.azsecure-data.org/other-data.html`

novos modelos de aprendizado de máquina, uma vez que a comparação é comprometida pela inviabilidade de reprodução dos experimentos existentes na literatura.

Como **recomendações**, destacamos que o detalhamento dos *datasets* deve incluir as fontes utilizadas, sejam estas repositórios de APKs ou *datasets* de terceiros. Além disso, é importante informar o *subset* utilizado no treinamento e validação dos modelos de aprendizado de máquina. Idealmente, recomendamos que sejam utilizadas fontes públicas para extrair as amostras, facilitando e acelerando a reprodução dos *datasets*. Complementarmente, a disponibilidade do conjunto exato de dados, utilizado no trabalho, viabilizaria uma reprodução fidedigna da pesquisa. É importante ressaltar também que devemos evitar fontes de dados antigas (*e.g.*, *datasets* de 2012 – a API do Android sofreu modificações significativas em 2015, por exemplo), pois não há garantias que os padrões encontrados pelos modelos preditivos, em amostras antigas, sejam aplicáveis em *malwares* atuais.

Dentre os **trabalhos futuros**, destacamos: analisar aspectos de reprodutibilidade dos modelos de aprendizado de máquina (*e.g.*, bibliotecas e hiperparâmetros utilizados).

Agradecimentos

Esta pesquisa foi financiada, conforme previsto nos Arts. 21 e 22 do decreto no. 10.521/2020, nos termos da Lei Federal no. 8.387/1991, através do convênio no. 003/2021, firmado entre ICOMP/UFAM, Flextronics da Amazônia Ltda e Motorola Mobility Comércio de Produtos Eletrônicos Ltda.

Referências

Alazab, M., Alazab, M., Shalaginov, A., Mesleh, A., and Awajan, A. (2020). Intelligent mobile malware detection using permission requests and api calls. *Future Generation Computer Systems*, 107:509–521.

Ali, M. A., Svetinovic, D., Aung, Z., and Lukman, S. (2017). Malware detection in android mobile platform using machine learning algorithms. In *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pages 763–768.

Amos, B., Turner, H., and White, J. (2013). Applying machine learning classifiers to dynamic android malware detection at scale. In *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1666–1671.

Arora, A., Peddoju, S. K., Chouhan, V., and Chaudhary, A. (2018). Hybrid android malware detection by combining supervised and unsupervised learning. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, page 798–800. ACM.

Arslan, R. S., Doğru, İ. A., and Barişçi, N. (2019). Permission-based malware detection system for android using machine learning techniques. *International journal of software engineering and knowledge engineering.*, 29(01):43–61.

Burguera, I., Zurutuza, U., and Nadjm-Tehrani, S. (2011). Crowdroid: Behavior-based malware detection system for android. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, page 15–26. ACM.

Cen, L., Gates, C. S., Si, L., and Li, N. (2015). A probabilistic discriminative model for android malware detection with decompiled source code. *IEEE Transactions on Dependable and Secure Computing*, 12(4):400–412.

Chawla, N., Kumar, H., and Mukhopadhyay, S. (2021). Machine learning in wavelet domain for electromagnetic emission based malware analysis. *IEEE Transactions on Information Forensics and Security*, 16:3426–3441.

Chen, X., Li, C., Wang, D., Wen, S., Zhang, J., Nepal, S., Xiang, Y., and Ren, K. (2020). Android hiv: A study of repackaging malware for evading machine-learning detection. *IEEE Transactions on Information Forensics and Security*, 15:987–1001.

Chen, Z., Yan, Q., Han, H., Wang, S., Peng, L., Wang, L., and Yang, B. (2018). Machine learning based mobile malware detection using highly imbalanced network traffic. *Information Sciences*, 433-434:346–364.

Demontis, A., Melis, M., Biggio, B., Maiorca, D., Arp, D., Rieck, K., Corona, I., Giacinto, G., and Roli, F. (2019). Yes, machine learning can be more secure! a case study on android malware detection. *IEEE Transactions on Dependable and Secure Computing*, 16(4):711–724.

Fan, M., Liu, J., Wang, W., Li, H., Tian, Z., and Liu, T. (2017). Dapasa: Detecting android piggybacked apps through sensitive subgraph analysis. *IEEE Transactions on Information Forensics and Security*, 12(8):1772–1785.

Ferrante, A., Malek, M., Martinelli, F., Mercaldo, F., and Milosevic, J. (2018). Extinguishing ransomware - a hybrid approach to android ransomware detection. In Imine, A., Fernandez, J. M., Marion, J.-Y., Logrippo, L., and Garcia-Alfaro, J., editors, *Foundations and Practice of Security*, pages 242–258, Cham. Springer International Publishing.

Gates, C. S., Li, N., Peng, H., Sarma, B., Qi, Y., Potharaju, R., Nita-Rotaru, C., and Molloy, I. (2014). Generating summary risk scores for mobile applications. *IEEE Transactions on Dependable and Secure Computing*, 11(3):238–251.

Jordaney, R., Sharad, K., Dash, S. K., Wang, Z., Papini, D., Nouretdinov, I., and Cavallaro, L. (2017). Transcend: Detecting concept drift in malware classification models. In *26th USENIX Security Symposium*, pages 625–642. USENIX Association.

Jung, J., Kim, H., Shin, D., Lee, M., Lee, H., Cho, S.-j., and Suh, K. (2018). Android malware detection based on useful api calls and machine learning. In *IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 175–178.

Kabakus, A. T. and Dogru, I. A. (2018). An in-depth analysis of android malware using hybrid techniques. *Digital Investigation*, 24:25–33.

Kouliaridis, V., Kambourakis, G., and Peng, T. (2020). Feature importance in android malware detection. In *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1449–1454.

Kumars, R., Alazab, M., and Wang, W. (2021). *A Survey of Intelligent Techniques for Android Malware Detection*, pages 121–162. Springer International Publishing, Cham.

Li, C., Chen, X., Wang, D., Wen, S., Ahmed, M. E., Camtepe, S., and Xiang, Y. (2021). Backdoor attack on machine learning based android malware detectors. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1.

Li, J., Sun, L., Yan, Q., Li, Z., Srisa-an, W., and Ye, H. (2018). Significant permission identification for machine-learning-based android malware detection. *IEEE Transactions on Industrial Informatics*, 14(7):3216–3225.

Ma, Z., Ge, H., Liu, Y., Zhao, M., and Ma, J. (2019). A combination method for android malware detection based on control flow graphs and machine learning algorithms. *IEEE Access*, 7:21235–21245.

Mahindru, A. and Sangal, A. L. (2021). MLDroid—framework for Android malware detection using machine learning techniques. *Neural Computing and Applications*, 33(10):5183–5240.

Mahindru, A. and Singh, P. (2017). Dynamic permissions based android malware detection using machine learning techniques. In *Proceedings of the 10th Innovations in Software Engineering Conference*, page 202–210. ACM.

Mas'ud, M. Z., Sahib, S., Abdollah, M. F., Selamat, S. R., and Yusof, R. (2014). Analysis of features selection and machine learning classifier in android malware detection. In *International Conference on Information Science Applications*, pages 1–5.

Narudin, F. A., Feizollah, A., Anuar, N. B., and Gani, A. (2016). Evaluation of machine learning classifiers for mobile malware detection. *Soft Computing*, 20(1):343–357.

Patel, K. and Buddadev, B. (2015). Detection and mitigation of android malware through hybrid approach. In Abawajy, J. H., Mukherjea, S., Thampi, S. M., and Ruiz-Martínez, A., editors, *Security in Computing and Communications*, pages 455–463, Cham. Springer International Publishing.

Peiravian, N. and Zhu, X. (2013). Machine learning for android malware detection using permission and api calls. In *IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 300–305.

Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., and Cavallaro, L. (2019). TESSERACT: Eliminating experimental bias in malware classification across space and time. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 729–746, Santa Clara, CA. USENIX Association.

Sahs, J. and Khan, L. (2012). A machine learning approach to android malware detection. In *European Intelligence and Security Informatics Conference*, pages 141–147.

Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., and Weiss, Y. (2012). "Andromaly": a behavioral malware detection framework for android devices. *Journal of Intelligent Information Systems*, 38(1):161–190.

Sharma, T. and Rattan, D. (2021). Malicious application detection in android — a systematic literature review. *Computer Science Review*, 40:100373.

Soares, T., Siqueira, G., Barcellos, L., Sayyed, R., Vargas, L., Rodrigues, G., Assolin, J., Pontes, J., Feitosa, E., and Kreutz, D. (2021). Detecção de malwares android: datasets e reprodutibilidade. `https://arxiv.kreutz.xyz/wrseg2021reprodutibilidade_ve1.pdf`.

Vinod, P., Zemmari, A., and Conti, M. (2019). A machine learning based approach to detect malicious android apps using discriminant system calls. *Future Generation Computer Systems.*, 94:333–350.

Wang, S., Chen, Z., Yan, Q., Yang, B., Peng, L., and Jia, Z. (2019). A mobile malware detection method using behavior features in network traffic. *Journal of Network and Computer Applications*, 133:15–25.

Wu, W.-C. and Hung, S.-H. (2014). Droiddolphin: A dynamic android malware detection framework using big data and machine learning. In *Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems*, page 247–252. ACM.

Xu, K., Li, Y., and Deng, R. H. (2016). Iccdetector: Icc-based malware detection on android. *IEEE Transactions on Information Forensics and Security*, 11(6):1252–1264.

Yerima, S. Y., Sezer, S., and Muttik, I. (2014). Android malware detection using parallel machine learning classifiers. In *Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, pages 37–42.

Yuan, Z., Lu, Y., Wang, Z., and Xue, Y. (2014). Droid-sec: Deep learning in android malware detection. *SIGCOMM Comput. Commun. Rev.*, 44(4):371–372.

Yuan, Z., Lu, Y., and Xue, Y. (2016). Droiddetector: android malware characterization and detection using deep learning. *Tsinghua Science and Technology*, 21(1):114–123.

Zarni Aung, W. Z. (2013). Permission-based android malware detection. *International Journal of Scientific & Technology Research*, 2(3):228–234.

Zhu, H.-J., You, Z.-H., Zhu, Z.-X., Shi, W.-L., Chen, X., and Cheng, L. (2018). Droiddet: Effective and robust detection of android malware using static analysis along with rotation forest model. *Neurocomputing*, 272:638–646.