XXI Escola Regional de Redes de Computadores / VIII Workshop de Segurança (ERRC/WRSeg). DOI: 10.5753/errc.2024.4688

Artigo Completo/Full Paper

Viés Avaliativo e Generalização Comprometida: O Impacto de Amostras Idênticas em Datasets de Malware Android

Evaluative Bias and Compromised Generalization: The Impact of Identical Samples in Android Malware Datasets

Gabriel Sousa Canto • gabrielcanto.canto6@gmail.com
Universidade Federal do Amazonas (UFAM)
Vanderson Rocha • vanderson@ufam.edu.br
Universidade Federal do Amazonas (UFAM)
Diego Kreutz • diegokreutz@unipampa.edu.br
Universidade Federal do Pampa (UNIPAMPA)
Hendrio Bragança • hendrio.luis@icomp.ufam.edu.br
Universidade Federal do Amazonas (UFAM)
Eduardo Feitosa • efeitosa@icomp.ufam.edu.br
Universidade Federal do Amazonas (UFAM)

Resumo. Neste trabalho, analisamos *datasets* públicos utilizados na detecção de *malwares* Android, investigando como amostras idênticas e os grupos que elas formam impactam o desempenho e a capacidade de generalização dos modelos de aprendizado de máquina. Nossos testes em seis cenários mostram que amostras idênticas elevam artificialmente as métricas de desempenho, criando uma impressão equivocada de eficácia. Além disso, em conjuntos com poucas amostras únicas, observamos que os modelos enfrentam dificuldades para generalizar em novos dados. Concluímos que é fundamental garantir amostras exclusivas no conjunto de testes para avaliações precisas e evitar conclusões enganosas sobre a capacidade dos classificadores.

Abstract. In this work, we analyzed public datasets used for Android malware detection, investigating how identical samples and the groups they form impact the performance and generalization ability of machine learning models. Our tests across six scenarios show that identical samples artificially inflate performance metrics, creating a misleading impression of efficacy. Additionally, in datasets with few unique samples, we observed that models struggle to generalize to new data. We conclude that ensuring exclusive samples in the test set is essential for accurate evaluations and to avoid misleading conclusions about classifier capabilities.

Palavras-chave: *Malware Android* • Aprendizado de Máquina • Amostras Idênticas • Amostras Únicas • Avaliação de Desempenho • Generalização

Keywords: Android Malware • Machine Learning • Identical Samples • Unique Samples • Performance Evaluation • Generalization

1 Introdução

A segurança de sistemas móveis, especialmente no contexto do sistema operacional Android, tornou-se um tema de crescente relevância devido ao uso intensivo de dispositivos móveis em várias áreas do cotidiano. Ferramentas automatizadas de análise de segurança, como classificadores de *malwares*, têm se mostrado essenciais para identificar comportamentos maliciosos e mitigar riscos. No entanto, a eficácia dessas ferramentas depende diretamente da qualidade dos dados nos quais elas são treinadas e avaliadas [1, 2].

Um dos problemas mais comuns em *datasets* para detecção de *malwares* Android é a presença de dados redundantes, que não agregam valor ao treinamento do modelo de aprendizado de máquina. A remoção de duplicatas é um passo essencial em pipelines de aprendizado de máquina, pois ajuda a evitar o sobreajuste [3] e garante que os modelos tenham uma boa capacidade de generalização. Quando amostras duplicadas no conjunto de treinamento aparecem no conjunto de testes, a avaliação é comprometida, pois estamos validando a capacidade de generalização do modelo com dados já vistos no treinamento. Além disso, se o conjunto de treinamento contiver muitas amostras duplicadas, a habilidade do modelo de generalizar será prejudicada [4, 5]. A presença de amostras duplicadas pode ainda afetar modelos de aprendizado de máquina ao inflar métricas de desempenho, distorcer a validação, introduzir viés na importância das características, desperdiçar recur-

sos computacionais, aumentar o risco de sobreajuste e agravar o desbalanceamento de classes [6–8, 4, 9].

Em 2019, estudos sobre duplicação de código em *Big Code* mostraram que métricas de desempenho podem ser artificialmente infladas em até 100% quando há dados duplicados, comprometendo a validade das conclusões em experimentos [7]. Em 2020, análises em *datasets* como CIFAR-10 e CIFAR-100 identificaram 3,3% e 10% de duplicatas nos conjuntos de teste em relação ao treinamento, e a substituição dessas amostras reduziu o desempenho de classificação em 9% a 14% [4].

Na detecção de *malwares* Android, o problema é agravado devido à alta incidência de amostras idênticas, resultante do uso de características binárias e limitadas a atributos genéricos. Frequentemente, aplicativos benignos e maliciosos compartilham características semelhantes, diferenciando-se apenas pela classificação. Essa duplicação pode induzir o modelo a superestimar sua generalização, fazendo-o aprender padrões específicos que não refletem a diversidade real dos dados, prejudicando o desempenho em novos exemplos.

Em [10], são reportadas métricas de *F-measure* entre 0,90 e 0,97 em um *dataset* com 43 permissões e 2 mil amostras, sem considerar a presença de amostras duplicadas. De forma similar, outros estudos [11, 12] apresentam métricas de desempenho excepcionalmente altas e propõem soluções que, à primeira vista, aparentam resolver o problema da detecção de *malwares*. No entanto, como evidenciado em nossa análise, muitos dos *datasets* amplamente utilizados na literatura apresentam uma quantidade significativa de amostras idênticas. Esse fator distorce as avaliações de desempenho e levanta questões sobre a real capacidade de generalização dos modelos propostos.

O objetivo deste trabalho é conduzir uma análise exploratória em *datasets* públicos amplamente utilizados na literatura, com o intuito de identificar e quantificar o número de amostras idênticas presentes. Além disso, buscamos avaliar o impacto dessas amostras no desempenho dos classificadores, investigando como a presença de dados idênticos pode influenciar os resultados dos modelos de aprendizado de máquina, especialmente no que se refere à sua capacidade de generalização.

2 Metodologia e Ambiente

Neste estudo, utilizamos *datasets* tabulares públicos disponíveis na literatura, contendo informações sobre a origem dos dados, nomes e tipos de características.

Na análise exploratória, identificamos e quantificamos amostras idênticas por meio de uma verificação sistemática, comparando cada amostra com as demais para detectar correspondências exatas (considerando apenas as características, sem a classificação). Em seguida, contabilizamos as classes de cada amostra nos grupos de amostras idênticas.

A Tabela 1 resume os *datasets* utilizados, detalhando suas características, o total de amostras e a distribuição entre amostras únicas e idênticas.

As características presentes nos *datasets* são predominantemente estáticas, ou seja, atributos que não mudam durante a execução do aplicativo e podem ser extraídos diretamente do código ou configuração do aplicativo [19]. Elas incluem permissões, que são privilégios solicitados pelos aplicativos para acessar funcionalidades do sistema; chamadas de API, que representam interações entre sistemas para fornecer serviços ou dados; intenções, mensagens assíncronas que permitem a comunicação entre componentes de diferentes aplicativos; e comandos do sistema, que são diretrizes extraídas dos arquivos .smali dos APKs para executar tarefas específicas.

Foram realizados cinco testes para avaliar o impacto das amostras idênticas no desempenho dos classificadores treinados:

Teste 1: Todas as amostras idênticas foram excluídas do conjunto de dados. *Objetivo:* Avaliar o impacto da remoção completa de amostras duplicadas na capacidade de generalização dos modelos.

Teste 2: Remoção do maior grupo de amostras idênticas, ou seja, instâncias com as mesmas características, independentemente de pertencerem à mesma classe ou não. *Objetivo:* Analisar como a remoção de um grupo significativo de amostras idênticas afeta a performance dos modelos, utilizando o restante dos dados para treinamento.

Teste 3: Conjunto de teste composto exclusivamente por amostras únicas. *Objetivo:* Avaliar a eficácia dos modelos treinados com dados contendo amostras idênticas ao serem testados exclusivamente com amostras distintas.

Teste 4: Utilização de conjuntos de dados sem amostras idênticas, onde, para cada grupo, foi mantida apenas uma instância. Esse teste foi realizado em dois cenários:

- **Teste 4.1:** A classe foi definida como benigna.
- **Teste 4.2:** A classe foi definida como *malware*.

Objetivo: Avaliar o comportamento dos classificadores em cenários sem dados sobrepostos, com uma única amostra representando cada grupo idêntico.

Teste 5: Conjunto de dados completo, com todas as amostras idênticas presentes. *Objetivo:* Servir como referência comparativa para os outros testes, mostrando

Tabela 1. *Resumo de informações dos datasets.*

Referência	*Dataset*	Características		A. únicas		A. idênticas		Total
		Qtde.	Tipos	Mal.	Ben.	Mal.	Ben.	
[13]	Adroit	166	P	133	992	3285	7066	11476
[14]	AndroCrawl	141	A(26), I(8), P(84), O(23)	7436	44860	2734	41714	96744
[15]	Android Permissions	151	P	1443	976	16344	8101	26864
[16]	DefenseDroid PRS	2877	P(1489), I(1388)	4620	2595	1380	3380	11975
[17]	Drebin-215	215	A(73), P(113), S(6), I(23)	1116	3870	4439	5606	15031
[18]	KronoDroid Emulator	276	P(145), A(123), O(8)	8401	14136	20344	21110	63991
[18]	KronoDroid Real	286	P(146), A(100), O(40)	14555	16908	26827	19847	78137

[P] Permissões, [A] Chamadas de API, [I] Intenções, [S] Comandos do Sistema, [O] Outros

Fonte: Elaborado pelos autores.

o impacto das amostras duplicadas no desempenho dos modelos.

Esses testes oferecem uma análise detalhada do impacto das amostras idênticas na capacidade de generalização dos classificadores.

Para os experimentos, selecionamos três classificadores que representam diferentes abordagens de aprendizado de máquina: *Support Vector Machine* (SVM), um modelo baseado em margens máximas; *Random Forest* (RF), um método de ensemble com árvores de decisão; e *K-Nearest Neighbors* (KNN), um classificador de instância baseado em proximidade. Essa diversidade permite uma análise abrangente do efeito das amostras idênticas em diferentes tipos de algoritmos.

Utilizamos a técnica de *holdout* com uma divisão de 80% para treino e 20% para teste, avaliando a capacidade de generalização dos modelos em dados não vistos. As métricas de desempenho adotadas foram precisão, *recall* [20, 21] e o Coeficiente de Correlação de Matthews (MCC) [22, 23].

Os testes foram conduzidos em uma máquina Dell Latitude 5420, equipada com 32 GB de RAM e um processador Intel® Core™ i7-1185G7 de 11ª geração, com *clock* de 3,00 GHz e 8 núcleos. O sistema operacional utilizado foi o Ubuntu 22.04.5 LTS, e a versão do Python utilizada foi a 3.10.

3 Resultados

As Tabelas 2, 3 e 4 apresentam os resultados obtidos para os *datasets* Adroit, Drebin-215 e DefenseDroid PRS. A análise identificou a formação de três grupos com resultados semelhantes; assim, optamos por apresentar os dados de um *dataset* representativo de cada grupo. Os resultados completos de todos os *datasets* estão publicamente disponíveis no GitHub[1].

O primeiro grupo é composto pelos *datasets* Adroit e Android Permissions, que se destacam pelo baixo desempenho, mesmo quando analisados em sua totalidade. Esse grupo apresenta uma alta concentração de amostras idênticas, o que impacta negativamente as métricas de desempenho, especialmente quando comparadas aos resultados do teste com o *dataset* completo (Teste 5).

O segundo grupo inclui os *datasets* Drebin-215, KronoDroid Real e KronoDroid Emulator. Esses conjuntos de dados exibem métricas elevadas (quase perfeitas) quando os classificadores são treinados com o *dataset* completo (Teste 5), mas sofrem reduções significativas no desempenho durante o Teste 3, que avalia o impacto da exclusão de amostras idênticas.

Por fim, o terceiro grupo é formado pelos *datasets* DefenseDroid PRS e Androcrawl. Nesse grupo, a comparação entre o Teste 5 e o Teste 3 revela um impacto mínimo nas métricas, com diferenças inferiores a 0,03. Em alguns casos, observa-se até um leve aumento no desempenho dos classificadores, indicando maior robustez desses *datasets* diante da remoção de amostras idênticas.

Ao comparar os Testes 3 e 5, fica evidente o impacto significativo das amostras idênticas quando não se garante que o conjunto de teste contenha apenas amostras únicas. No Teste 5, os resultados foram artificialmente inflacionados em todos os *datasets*, apresentando um desempenho ilusório.

O *dataset* Adroit foi particularmente afetado, com reduções expressivas conforme mostrado na Tabela 2. Houve quedas acentuadas, superiores a 0,1, nas métricas de precisão, *recall* e MCC. Isso se deve ao fato

[1] https://github.com/Malware-Hunter/WRSeg24-ViesAvaliativo

Tabela 2. Resultados Adroit

Teste	Modelo	Precisão	*Recall*	MCC
Teste 1	SVM	1,0000	0,0167	0,1174
	RF	0,1143	0,0667	-0,0548
	KNN	0,2647	0,1500	0,0776
Teste 2	SVM	0,3030	0,1504	0,1429
	RF	0,3077	0,1203	0,1292
	KNN	0,1707	0,1579	0,0570
Teste 3	SVM	0,4248	0,1611	0,1753
	RF	0,5000	0,1946	0,2313
	KNN	0,3393	0,1913	0,1431
Teste 4.1	SVM	0,0000	0,0000	0,0000
	RF	0,1667	0,0323	0,0322
	KNN	0,0000	0,0000	-0,0457
Teste 4.2	SVM	0,5000	0,3043	0,1519
	RF	0,4111	0,3217	0,0633
	KNN	0,3395	0,4783	-0,0495
Teste 5	SVM	0,9512	0,7131	0,7629
	RF	0,9346	0,7834	0,8003
	KNN	0,8700	0,8003	0,7649

Fonte: Elaborado pelos autores.

Tabela 3. Resultados Drebin

Teste	Modelo	Precisão	*Recall*	MCC
Teste 1	SVM	0,9754	0,9215	0,9327
	RF	0,9749	0,9041	0,9210
	KNN	0,9599	0,9041	0,9113
Teste 2	SVM	0.9689	0.9467	0.9102
	RF	0.9576	0.888	0.9008
	KNN	0.8962	0.9017	0.8694
Teste 3	SVM	0,9534	0,9367	0,9272
	RF	0,9703	0,9259	0,9314
	KNN	0,9197	0,9259	0,8974
Teste 4.1	SVM	0,9784	0,6099	0,7443
	RF	0,9836	0,5381	0,6972
	KNN	0,9044	0,5516	0,6694
Teste 4.2	SVM	0,8623	0,9439	0,8105
	RF	0,7824	0,9453	0,7196
	KNN	0,7313	0,9365	0,6464
Teste 5	SVM	0,9905	0,9694	0,9689
	RF	0,9944	0,9851	0,9841
	KNN	0,9758	0,9740	0,9609

Fonte: Elaborado pelos autores.

de que o *dataset* é predominantemente composto por amostras idênticas, totalizando 10.351 duplicatas, o que representa 90,2% do total de amostras, conforme Tabela 1.

No Drebin-215, as reduções em precisão e *recall* variaram entre 0,02 e 0,06, enquanto as quedas no MCC foram menos significativas. Embora tenha ocorrido uma redução nas métricas, os valores ainda se mantêm elevados, com mais de 0,9 em ambos os casos.

Já o *dataset* DefenseDroid PRS apresentou um comportamento distinto, com aumento nas métricas, exceto por reduções observadas no *recall* do RF e do KNN.

Tabela 4. Resultados DefenseDroid PRS

Teste	Modelo	Precisão	*Recall*	MCC
Teste 1	SVM	0,9250	0,9063	0,8059
	RF	0,9208	0,9013	0,7954
	KNN	0,9059	0,9013	0,7758
Teste 2	SVM	0,9457	0,8918	0,8393
	RF	0,9306	0,8993	0,8297
	KNN	0,9088	0,8775	0,7862
Teste 3	SVM	0,9620	0,9033	0,8484
	RF	0,9598	0,9025	0,8448
	KNN	0,9402	0,8924	0,8102
Teste 4.1	SVM	0,9365	0,9062	0,8337
	RF	0,9327	0,8664	0,7921
	KNN	0,9276	0,8416	0,7639
Teste 4.2	SVM	0,8745	0,9578	0,6928
	RF	0,8606	0,9553	0,6570
	KNN	0,8241	0,9619	0,5762
Teste 5	SVM	0,9404	0,8898	0,8358
	RF	0,9294	0,9075	0,8399
	KNN	0,8945	0,9058	0,8005

Fonte: Elaborado pelos autores.

Esse desempenho está possivelmente relacionado ao fato de o DefenseDroid PRS possuir a menor proporção de amostras idênticas (4.760, ou 39,8% do total), o que reflete o grande número de características presentes nesse *dataset*.

O Teste 1, que simula um cenário extremo onde todas as amostras idênticas são removidas, resultou em uma queda geral no desempenho dos modelos quando comparado ao Teste 3. A exclusão completa dessas amostras não é uma solução viável, pois a grande quantidade de dados eliminados prejudica diretamente o desempenho dos classificadores.

No Teste 2, que avaliou a exclusão do maior grupo de amostras idênticas, os resultados mostraram tanto aumentos quanto reduções no desempenho, levando a um comportamento semelhante ao do Teste 1, sem diferenças significativas nos impactos causados.

Nos Testes 4.1 e 4.2, que eliminaram amostras sobrepostas, observou-se que a exclusão de grandes quantidades de dados prejudicou o desempenho dos classificadores em todos os cenários. O *dataset* Drebin-215, por exemplo, registrou quedas de precisão abaixo de 0,6 no Teste 4.1, enquanto o DefenseDroid PRS foi o menos afetado, mantendo métricas acima de 0,8. Esses testes demonstram que remover amostras sobrepostas afeta negativamente os resultados, sem evidências de que a exclusão melhora o desempenho geral.

Os resultados dos Testes 1 e 2 indicam que a remoção parcial ou total de amostras idênticas prejudica o desempenho dos modelos. No entanto, *datasets* com alta concentração de duplicatas, como Adroit e Android

Permissions, apresentam baixa capacidade de generalização para novos dados, resultando em desempenho insatisfatório em todos os cenários.

Para o *dataset* Adroit, os classificadores demonstraram dificuldades significativas para classificar corretamente as amostras de *malwares*. O único cenário em que as métricas apresentaram resultados satisfatórios foi ao utilizar o *dataset* completo, o que sugere que os classificadores estão enviesados pela alta quantidade de amostras idênticas, conforme ilustrado na Tabela 1. Esses resultados indicam que o *dataset* Adroit possui baixa qualidade nas suas amostras e características.

O *dataset* Android Permissions apresentou problemas semelhantes aos do Adroit, com os classificadores mostrando desempenho insatisfatório nas métricas de acurácia, precisão e MCC. A baixa precisão foi particularmente impactada pelo desbalanceamento dos dados, já que o número reduzido de aplicativos benignos comprometeu a capacidade dos modelos de identificar corretamente essa classe.

Os resultados indicam que a presença de amostras idênticas pode mascarar a verdadeira capacidade de generalização dos classificadores. Embora o uso de todas as amostras (Teste 5) produza métricas elevadas, isso pode sugerir um *overfitting* oculto. A exclusão parcial ou total de amostras duplicadas (Testes 1 e 2) revela uma leve queda na performance, mas demonstra que os modelos são mais propensos a generalizar para dados inéditos, como indicado pelos resultados do Teste 3.

Além disso, a estratégia de manter uma única amostra por grupo (Testes 4.1 e 4.2) mostrou que a definição da classe tem pouco impacto no desempenho final dos modelos, sugerindo que dados sobrepostos não comprometem significativamente o desempenho para os *datasets* analisados.

4 Conclusão

A comparação entre os Testes 3 e 5 mostrou que a presença de amostras idênticas nos *datasets* afeta significativamente o desempenho dos modelos de classificação. Quando o conjunto de teste inclui amostras duplicadas, as métricas de desempenho são artificialmente inflacionadas, criando uma falsa impressão de alta performance.

No Teste 1, a remoção total das amostras idênticas resultou, em geral, na redução das métricas de desempenho, indicando que os classificadores foram impactados pela exclusão dessas amostras. Embora alguns casos tenham mostrado um aumento pontual na precisão (como no SVM com o *dataset* Adroit), a tendência foi de queda no *recall* e na precisão em diversos cenários. O Teste 2 apresentou um padrão semelhante, mas com impactos menos acentuados.

Nos cenários 4.1 e 4.2, observou-se que amostras idênticas com sobreposição de classes não prejudicam os modelos treinados e que manipulá-las pode, na verdade, afetar negativamente o desempenho. No Teste 4.1, onde amostras benignas foram preservadas, a precisão aumentou levemente, mas o *recall* foi prejudicado. No Teste 4.2, mantendo amostras maliciosas, o comportamento foi oposto, com queda na precisão e leve aumento no *recall*.

Esses resultados indicam que o desempenho dos classificadores é frequentemente superestimado devido à presença de amostras idênticas, um problema comum nos *datasets* de *malwares*. Os experimentos reforçam a importância de garantir que o conjunto de teste contenha apenas amostras únicas e que o *dataset* tenha uma quantidade adequada de amostras exclusivas. Caso contrário, a alta prevalência de duplicatas — como nos *datasets* Adroit e Android Permissions — compromete a capacidade de generalização dos classificadores, resultando em uma performance artificialmente elevada que não se sustenta em ambientes reais.

Financiamento

Esta pesquisa foi parcialmente 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. O presente trabalho foi realizado também com apoio da CAPES – Código de Financiamento 001 e da FAPERGS, através dos termos de outorga 24/2551-0001368-7 e 24/2551-0000726-1.

Referências

1 Zhao, Y. *et al.* On the Impact of Sample Duplication in Machine-Learning-Based Android Malware Detection. *ACM Trans. Softw. Eng. Methodol.*, Association for Computing Machinery, New York, NY, USA, v. 30, n. 3, mai. 2021. ISSN 1049-331X. DOI: 10.1145/3446905. Disponível em: https://doi.org/10.1145/3446905.

2 Gaber, M. G.; Ahmed, M.; Janicke, H. Malware detection with artificial intelligence: A systematic literature review. *ACM Computing Surveys*, ACM New York, NY, v. 56, n. 6, p. 1–33, 2024.

3 Budach, L. *et al.* The effects of data quality on machine learning performance. *arXiv preprint arXiv:2207.14529*, 2022.

4 Barz, B.; Denzler, J. Do we train on test data? purging cifar of near-duplicates. *Journal of Imaging*, MDPI, v. 6, n. 6, p. 41, 2020.

5 Sarracino, F.; Mikucka, M. Estimation bias due to duplicated observations: a Monte Carlo simulation, 2016.

6 Huang, K. *et al.* Learning classifiers from imbalanced data based on biased minimax probability machine. *In*: IEEE. PROCEEDINGS of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. 2004. v. 2, p. ii–ii.

7 Allamanis, M. The adverse effects of code duplication in machine learning models of code. *In*: PROCEEDINGS of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software. 2019. P. 143–153.

8 Pendlebury, F. *et al.* {TESSERACT}: Eliminating experimental bias in malware classification across space and time. *In*: 28TH USENIX security symposium (USENIX Security 19). 2019. P. 729–746.

9 Alam, M. T.; Bhusal, D.; Rastogi, N. Revisiting Static Feature-Based Android Malware Detection. *arXiv preprint arXiv:2409.07397*, 2024.

10 Şahin, D. Ö. *et al.* A novel permission-based Android malware detection system using feature selection based on linear regression. *Neural Computing and Applications*, Springer, p. 1–16, 2023.

11 Mathur, A. *et al.* NATICUSdroid: A malware detection framework for Android using native and custom permissions. *Journal of Information Security and Applications*, Elsevier, v. 58, p. 102696, 2021.

12 Palumbo, P. *et al.* A pragmatic android malware detection procedure. *Computers & Security*, Elsevier, v. 70, p. 689–701, 2017.

13 Martín, A. *et al.* ADROIT: Android malware detection using meta-information. *In*: IEEE. 2016 IEEE Symposium Series on Computational Intelligence (SSCI). 2016. P. 1–8.

14 Sisto, A. AndroCrawl: studying alternative Android marketplaces. Politecnico di Milano, 2012.

15 Mahindru, A. Android permission dataset. *Mendeley Data*, v. 1, p. 2018, 2018.

16 Colaco, C. *et al.* Defensedroid: A modern approach to android malware detection. *Strad Research*, v. 8, n. 5, p. 271–282, 2021.

17 Yerima, S. Y.; Sezer, S. Droidfusion: A novel multilevel classifier fusion approach for android malware detection. *IEEE transactions on cybernetics*, IEEE, v. 49, n. 2, p. 453–466, 2018.

18 Guerra-Manzanares, A.; Bahsi, H.; Nõmm, S. Kronodroid: time-based hybrid-featured dataset for effective android malware detection and characterization. *Computers & Security*, Elsevier, v. 110, p. 102399, 2021.

19 Wang, W. *et al.* Constructing features for detecting android malicious applications: issues, taxonomy and directions. *IEEE access*, IEEE, v. 7, p. 67602–67631, 2019.

20 Powers, D. M. W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011.

21 Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, Elsevier, v. 45, n. 4, p. 427–437, 2009.

22 Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, Elsevier, v. 405, n. 2, p. 442–451, 1975.

23 Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, BioMed Central, v. 21, n. 1, p. 6, 2020.