MalSynGen: redes neurais artificiais na geração de dados tabulares sintéticos para detecção de *malware*

Angelo Gaspar Diniz Nogueira[1], Kayua Oleques Paim[2], Hendrio Bragança[3], Rodrigo Mansilha[1], Diego Kreutz[1]

[1]LEA e PPGES, Universidade Federal do Pampa (UNIPAMPA) – Alegrete, Brasil

[2]IComp, Universidade Federal do Amazonas (UFAM) – Manaus, Brasil

[3]IC, Universidade Federal do Rio Grande do Sul (UFRGS) – Porto Alegre, Brasil

Resumo. *A MalSynGen é uma ferramenta que utiliza redes neurais artificiais para gerar dados sintéticos tabulares para o domínio de malware Android. Para avaliar sua performance foram aumentados os dados de dois datasets, considerando métricas de fidelidade estatística e utilidade. Os resultados indicam que MalSynGen é capaz de capturar padrões representativos para o aumento de dados tabulares.*

1. Introdução

Diversas técnicas de aprendizado de máquina têm se mostrado promissoras e efetivas para mitigar o crescente problema de *malware* [Brown et al., 2024]. Porém, a eficiência desses métodos está intimamente relacionada à quantidade de amostras (i.e., número suficientemente representativo de amostras), representatividade de atributos (i.e., tipos e quantidades de características) e ao frescor (i.e., atualidade das amostras em comparação com avanços tecnológicos) dos *datasets* utilizados no treinamento [Paullada, A. et. al., 2021].

Estudos recentes demonstram que a obtenção de quantidades significativas e representativas de dados atualizados é uma tarefa complexa, custosa e significativamente demorada [Rocha V. et. al, 2023]. Como forma de mitigar esse problema, começaram a surgir diferentes ferramentas para geração de dados sintéticos, como [Xu et al., 2019, Rajabi and Garibay, 2022]. Entretanto, técnicas para avaliação qualitativa de dados sintéticos ainda são consideradas um desafio em aberto [Platzer and Reutterer, 2021], isto é, não há recomendações específicas e nem consenso sobre o assunto.

Neste trabalho propomos a ferramenta MalSynGen (*Malware Syntetic data Generation*), cujo principal objetivo é gerar e avaliar dados sintéticos tabulares úteis contexto de detecção de *malware* Android. Para atingir esse objetivo, adotamos uma metodologia distinta das soluções existentes. A MalSynGen implementa uma metodologia sistemática de avaliação que incorpora métricas como erro quadrático, valor de p, e similaridade de cosseno (coletivamente referidas como *fidelidade* estatística), além das métricas de desempenho dos classificadores, utilizadas para definir a *utilidade* dos dados. É importante destacarmos também que dados tabulares ainda estão entre os mais frequentemente utilizados para o desenvolvimento

de modelos de aprendizado de máquina no contexto de detecção de *malware* Android [Kouliaridis and Kambourakis, 2021].

A MalSynGen utiliza arquiteturas de redes neurais artificiais do tipo GAN (*Generative Adversarial Network*) condicional (cGAN) para ampliar *datasets* e assim obter melhores classificadores e resultados. Para avaliar as arquiteturas de redes neurais, utilizamos dois conjuntos de dados distintos. Após experimentações para definir os hiperparâmetros adequados, conseguimos obter resultados que podem ser considerados positivos quando analisamos as métricas de *fidelidade* e *utilidade* para ambos os *datasets*.

Como contribuições do trabalho podemos destacar: **(a)** uma arquitetura de redes neurais artificiais para geração de dados tabulares sintéticos; **(b)** uma nova metodologia de treinamento e avaliação das redes generativas para o contexto de *malware* Android; **(c)** introdução de métricas específicas para avaliação de redes generativas, incluindo *fidelidade* estatística e *utilidade*, complementares às métricas já utilizadas na literatura.

2. Trabalhos relacionados

Na Tabela 1 apresentamos os principais trabalhos relacionados no contexto de aumento de dados tabulares. Listamos as técnicas, as métricas, o domínio e os *datasets* utilizados em cada trabalho. Como pode ser observado, as GANs são frequentemente utilizadas para geração de dados sintéticos tabulares.

Tabela 1. Trabalhos relacionados à geração de dados tabulares sintéticos.

Trabalho	Técnica	Métricas	Domínio	Dataset
[Xu and Veeramachaneni, 2018]	GAN	pontuação F1, acurácia, erro quadrático médio e absoluto	Dados gerais	Census-Income, KDD Cup 1999 e Covertype
[Park, N. et. al, 2018]	GAN	Distribuição cumulativa, pontuação F1, erro médio relativo, distancia euclidiana e AUC	Dados gerais	LACity, Adult, Health e Airline
[Xu et al., 2019]	cGAN	Acurácia, pontuação F1, R^2 e log de verossimilhança	Dados gerais	MNIST, adult, census, covertype, intrusion, credit e news
[Rajabi and Garibay, 2022]	GAN	Acurácia, F1 score e discrimination score	Dados demograficos	UCI Adult, Bank Marketing, ProPublica e LSAC
[Choi, E. et. al., 2017]	GAN	Precisão, pontuação F1, recall, distribuição de Bernoulli	Saúde	MIMIC-III, Falhas cardíacas sutter
[Mimura, 2020]	GAN	Acurácia, pontuação F1 e Recall	Malware VBA	Virus Total
[Casola, K. et. al., 2023]	cGAN	pontuação F1, recall, acurácia, precisão divergência KL, máxima discrepância média e erro médio Quadrático	Malware Android	Drebin 215
[Amin, M. et. al, 2022]	GAN	pontuação F1, recall, acurácia e precisão, AUC, FPR e cobertura	Malware Android	Drebin e AMD
[Li et al., 2024]	cGAN	pontuação F1, recall, acurácia e precisão	Malware Android	CCCS-CIC e Mal2020
Este trabalho	cGAN	pontuação F1, recall, acurácia, precisão, AUC, cosseno de similaridade, erro quadrático, valor de p e discrepância média	Malware Android	KronoDroid R e KronoDroid E

A maioria das soluções para dados tabulares procura observar as particularidades de algum contexto específico, como saúde [Choi, E. et. al., 2017], demografia [Rajabi and Garibay, 2022] e *malware* VBA [Mimura, 2020]. Podemos também observar que existem algumas soluções específicas ao contexto de *malware* Android, onde podemos constatar uma predominância das cGANs. No entanto, as soluções anteriores não consideram métricas para avaliar a *fidelidade* estatística dos dados sintetizados, apenas sua *utilidade*. Além disso, não fazem uso de uma validação cruzada na fase de validação.

Em termos de métricas, podemos observar que todos trabalhos utilizam métricas clássicas de classificação binária, como precisão, acurácia, *recall* e pontuação F1 para avaliar o desempenho dos dados sintéticos. Complementarmente, além das métricas mencionadas, nós introduzimos também métricas de *utilidade* e *fidelidade*, que são adequadas no processo de avaliação redes generativas e dados sintéticos, conforme apontado em pesquisas recentes [Canbek et al., 2021, Rainio et al., 2024].

Diferentemente dos demais trabalhos, nós não apenas disponibilizamos uma ferramenta pública voltada para aumento de dados tabulares, mas também apresentamos uma nova metodologia para avaliação sistemática da geração de dados sintéticos no contexto de *malware* Android. Em nossa metodologia adotamos uma comparação cruzada entre dobras (*folds*) de dois classificadores, similar ao proposto em [Esteban et al., 2017] (TRTS e TSTR): treinando um classificador com dados reais e avaliando ele com dados sintéticos; e treinando um classificador com dados sintéticos e avaliando ele com dados reais. Porém, em contraste a metodologia originalmente proposta, utilizamos métricas distintas em nossa avaliação (i.e., similaridade de cosseno), e realizamos uma análise estatística utilizando o teste de *Wilcoxon* para verificar se há alguma diferença estatisticamente significativa entre o desempenho dos classificadores. Finalmente, a MalSynGen incorpora também uma solução de rastreamento dos experimentos e visualização dos resultados.

3. MalSynGen: Processo e Implementação

Nesta seção apresentamos os processos e aspectos de implementação da MalSynGen.

3.1. Processo

Ilustramos o fluxo de execução da MalSynGen na Figura 1. O fluxo proposto é composto por três etapas principais: seleção e manipulação do *dataset*, treinamento da rede neural geradora (bem como classificadores) e avaliação de cGANs.

Na primeira etapa, de **seleção**, escolhemos um *dataset* e realizamos o balanceamento pela classe (benigna ou maligna) com menor quantidade de amostras. O balanceamento das amostras benignas e malignas do *dataset* é realizado através do uso de técnicas de subamostragem. Em seguida, preparamos o *dataset* para validação cruzada por meio de k-dobras (do inglês *k-folds*). O *dataset* balanceado é dividido em k subconjuntos de tamanhos iguais e, em cada iteração, uma parte é escolhida como subconjunto de avaliação (*Dataset* r) e os restantes *(k-1)* subconjuntos são usados para treinamento (*Dataset* R).

Na etapa de **treinamento**, a ferramenta recebe como entrada os hiperparâmetros de treinamento da cGAN[1] e hiperparâmetros dos algoritmos de classificação. Ademais, para cada dobra, são recebidos os respectivos subconjuntos de treinamento (R) e de avaliação (r). O *dataset* de treinamento (R) e parte dos hiperparâmetros são utilizados para treinar a rede neural geradora cGAN. A instância treinada da cGAN é utilizada para gerar um *dataset* sintético de treino (S). Em seguida, a mesma instância da cGAN é usada para gerar *dataset* sintético de avaliação

[1]Neste trabalho implementamos uma arquitetura própria de cGAN.

(s), transformando dados do *dataset* real de avaliação (r). É importante ressaltar que os dados de avaliação e os dados sintéticos de avaliação não são utilizados em nenhum momento para treinar ou ajustar o modelo generativo cGAN.

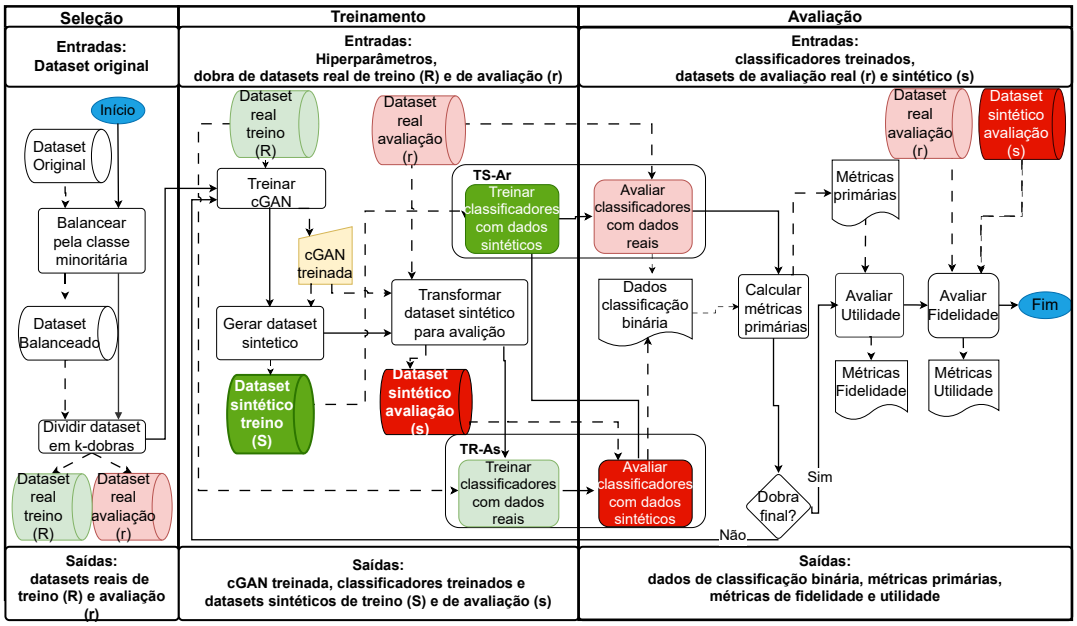

Figura 1. Seleção, treinamento e avaliação de cGANs usando a MalSynGen. Setas contínuas indicam a ordem dos processos e as setas pontilhadas indicam o sentido de criação ou utilização de artefatos (i.e., *datasets*, modelos, dados brutos e métricas de desempenho) nos processos. Alguns artefatos são representados múltiplas vezes para simplificar o cruzamento de setas.

Além de treinar e usar cGANs para gerar dados sintéticos, precisamos treinar classificadores para posteriormente verificarmos a *utilidade* dos dados sintéticos gerados. Implementamos os dois métodos propostos em [Esteban et al., 2017]: treinar com dados reais (R) e avaliar com dados sintéticos (s) (TR-As) (do inglês *Train on Real, Test on Synthetic* - TRTS), e; treinar com dados sintéticos (S) e avaliar com dados reais (r) (TS-Ar) (do inglês *Train on Synthetic, Test on Real* – TSTR).

No processo de **avaliação**, executamos os classificadores e calculamos métricas primárias, de *utilidade* e de *fidelidade*. Para cada instância (classificador treinado e correspondente *dataset* de avaliação), extraímos os resultados de testes classificação binária, a saber: quantidades de verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo. A partir dos resultados de classificação binária, geramos matrizes de confusão e calculamos métricas primárias comumente utilizadas em problemas de classificação binária, como Acurácia, Precisão, *Recall* e Pontuação F1. Como falsos negativos são considerados mais prejudiciais que falsos positivos no contexto de detecção de *malwares*, consideramos a métrica *Recall* como sendo mais relevante do que a *Precisão*.

Após a avaliação da dobra final, passamos para a avaliação de *utilidade*. Para isso, nós extraímos a média e o desvio padrão de cada métrica primária para cada classificador seguindo os dois métodos considerados (TS-Ar e TR-As).

Na sequência, realizamos uma avaliação de *fidelidade* para verificar se o *data-*

set sintético (s) reproduz as características estatísticas do *dataset* real (r). Com esse propósito, a ferramenta pode incluir diversas métricas como similaridade do cosseno e do erro quadrático p. Esses métodos consideram e avaliam não apenas as distribuições características individuais entre os *datasets*, mas também as correlações multivariáveis entre características.

Finalmente, verificamos se a *utilidade* dos dados sintéticos é fiel à *utilidade* dos dados reais através de testes como o de *Wilcoxon* [Wilcoxon, 1945]. O teste de *Wilcoxon* opera sobre duas amostras pareadas com duas hipóteses: H_0, que indica não existir diferença estatisticamente significativa entre as amostras, e H_1, que afirma que há uma diferença estaticamente significativa. Isso é determinado através do cálculo da diferença absoluta entre as amostras pareadas, seguido pelo *ranking* destas diferenças e cálculo de coeficiente de concordância. O resultado desse processo é um valor de p, que é comparado com um limiar *threshold* para rejeitar ou aceitar a hipótese H_0. Nós utilizamos a média calculada entre os valores de p resultantes das métricas de *utilidade* para estas hipóteses. A razão para seu uso é obter uma avaliação abrangente dos parâmetros dos classificadores.

3.2. Implementação e Experimentação

Para a implementação e execução da MalSynGen, utilizamos Python (versão 3.8) e as seguintes bibliotecas principais: *numpy* 1.21.5, *Keras* 2.9.0, *Tensorflow* 2.9.1, *pandas* 1.4.4, *scikit-learn* 1.1.1. e *mlflow*[2] 2.12.1. Utilizamos um servidor AMD Ryzen 7 5800x como processador de 8 cores e 64 GB de memória RAM para executar o experimentos. O sistema operacional do servidor é o Ubuntu Server versão 22.04. Detalhes da implementação da cGAN, recursos utilizados e resultados de todas as execuções podem ser vistos no repositório GitHub [Nogueira, A. et. al., 2024].

4. Avaliação

O objetivo da avaliação é demonstrar o uso da MalSynGen para gerar *datasets* sintéticos. Nesse sentido, são necessários meios para verificar (i) se o *dataset* sintético preserva as características essenciais do *dataset* original (isto é, são fiéis), ampliando-o e diversificando-o de forma efetiva; e (ii) se os dados sintéticos podem ser utilizados com classificadores (isto é, são úteis).

A Tabela 2 resume os valores dos hiperparâmetros utilizados na instanciação e treinamento do nosso modelo de cGAN. A avaliação foi realizada através da validação cruzada com 10 dobras (k-fold = 10) e tamanho de *batch* de 256.

Tabela 2. Hiperparâmetros utilizados nos experimentos.

Hiperparâmetro	Valor
Dense Layer Sizes (d)	2048
Dense Layer Sizes (g)	4096
Dropout Decay Rate (d)	0,4
Dropout Decay Rate (g)	0,2
Initializer Deviation	0,02
Initializer Mean	0
Latent Dimension	128

Hiperparâmetro	Valor
Latent Stander Deviation	1
Number Epochs	500
Optimizer Discriminator Learning	0,0001
Optimizer Generator Learning	0,0001
Conditional GAN ADA Delta Learning Rate	0,001
Conditional GAN RMS Prop Learning Rate	0,001
Conditional GAN ADAM Learning Rate	0,0001

[2]Biblioteca utilizada para o rastreamento detalhado da execução dos experimentos, incluindo geração e versionamento de artefatos.

Na Tabela 3 listamos as especificações dos *datasets* considerados neste estudo. Ambos *datasets* são frequentemente utilizados em pesquisas no contexto de detecção de *malware* Android e representam dados de dois conjuntos distintos de dispositivos, isto é, físicos (reais) e emuladores.

Tabela 3. Datasets utilizados na avaliação do MalSynGen.

Dataset	Características	Amostras			Características (balanceado)	N. Amostras (Balanceado)
		Malware	Benignos	Total		
KronoDroid E	482	44,90%	55,10%	63991	276	20000
KronoDroid R	483	53%	47%	78137	285	20000

4.1. Métricas de fidelidade

Na Tabela 4 apresentamos os resultados obtidos para as métricas de *fidelidade* (similaridade de cosseno e erro quadrático médio). Os resultados da métrica de similaridade de cosseno indicam que os *dados sintéticos são parecidos mas não são idênticos aos dados originais*. Os valores obtidos para o erro quadrático confirmam a alta similaridade, já que apresentam valores próximos a zero, mas não zero.

Tabela 4. Valores obtidos para métricas de Fidelidade.

Dataset	Positivo		Falso	
	Cosseno	Erro quadrático	Cosseno	Erro quadrático
KronoDroid E	0,77	0,10	0,77	0,07
KronoDroid R	0,71	0,11	0,73	0,07

4.2. Métricas de utilidade

Na Figura 2 mostramos as médias dos resultados obtidos após a avaliação de 10 dobras considerando o classificador *Random Forest*. São apresentados dois grupos de barras referentes, respectivamente, aos *datasets* Kronodroid E e Kronodroid R. Cada grupo contém 4 pares de barras: um par para cada métrica (acurácia, precisão, pontuação F1 e *recall*), onde uma barra mostra o valor para o modelo TS-Ar (barras claras) e a outra, para o modelo TR-As (barras escuras).

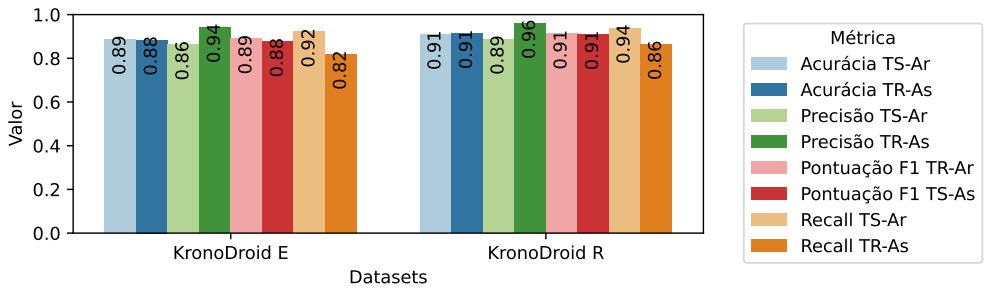

Figura 2. Resultados entre os *datasets* com o classificador *Random Forest*.

Podemos observar na Figura 2 que todos os resultados podem ser considerados bons pois são igual ou acima de 82%, chegando a 96%. É importante destacar que dentre essas métricas, a *Recall* é mais relevante pois falsos negativos são considerados mais prejudiciais que os falsos positivos.

Na Tabela 5 listamos os valores p obtidos através do teste de *Wilcoxon* para as métricas de acurácia, precisão, pontuação F1, *recall* e AUC da curva de ROC do modelo aprendizado de máquina – *Random Forest* – aplicados aos dois conjuntos de dados. Esses valores p indicam a significância estatística dos modelos em cada conjunto de dados. Para a realização dos experimentos foi adotado o valor de 0,05 como limiar de p, um valor amplamente aceito e utilizado na literatura.

Tabela 5. Valores de *p* das métricas do classificador *Random Forest*.

Dataset	P_value					
	Acurácia	**Precisão**	**Pontuação F1**	**Recall**	**Média**	**AUC**
KronoDroid E	0,770	0,002	0,064	0,002	0,209	0,375
KronoDroid R	0,922	0,002	0,846	0,008	0,444	0,625

A nossa hipótese H_0 é que não existe diferença estatisticamente significativa entre as métricas obtidas pelos classificadores TS-Ar e TR-As. Conforme observado, as médias dos valores de p dos classificadores foram superiores a 0,05 para as métricas de acurácia, pontuação F1 e AUC, mas inferiores em precisão e *recall*. Portanto, conclui-se que não há evidencias suficientes para rejeitar a hipótese H_0 para acurácia, pontuação F1, média e AUC. Esses resultados sugerem que os *datasets* sintéticos sejam mais representativos das características gerais dos dados reais.

Na Tabela 6 mostramos os valores p obtidos pelo teste de *Wilcoxon* para a média das métricas de utilidade para cada classificador considerado. Pode ser observado que todos os classificadores retornam uma média maior que o limiar de 0,05. Portanto, concluímos que não existe uma diferença significativa entre os *datasets*, independente do classificador considerado.

Tabela 6. Valor de *p* obtido por 6 classificadores.

Dataset	P_value					
	Random Forest	**Decision Tree**	**Adaboost**	**Perceptron**	**SVM**	**XGboost**
Kronodroid E	0,209	0,060	0,198	0,173	0,234	0,291
Kronodroid R	0,444	0,195	0,125	0,345	0,482	0,173

5. Considerações Finais

A MalSynGen é uma ferramenta publicamente disponível para treinamento e avaliação de redes geradoras cGANs para o processo de geração de *datasets* tabulares sintéticos no contexto de detecção de *malware* Android. Os resultados obtidos com os *datasets* KronoDroid demonstram que conjuntos de dados sintetizados pela MalSynGen podem ser considerados *úteis* para diversos classificadores e *fiéis* quanto às características gerais dos dados originais.

Como trabalhos futuros, podemos elencar: (a) incluir outras métricas de utilidade, fidelidade, desempenho computacional e novos classificadores; (b) ampliar o número de *datasets* utilizados no aumento de dados, para verificar a capacidade de generalização da ferramenta; (c) uma análise comparativa de desempenho com ferramentas similares utilizando os mesmos conjuntos de dados; (d) generalizar a ferramenta para incluir outros métodos de geração de dados tabulares sintéticos.

Finalmente, a **demonstração** da ferramenta será realizada em dispositivo próprio dos autores. Iremos realizar a demonstração de funcionamento da MalSyn-

Gen através dos seguintes passos: (a) apresentação das funcionalidades; (b) apresentação dos parâmetros de execução; (c) apresentação da execução e resultados para diferentes *datasets*.

Agradecimentos. A pesquisa contou com apoio parcial da RNP (Programa Hackers do Bem - GT Malware DataLab), da CAPES (Código de Financiamento 001) e da FAPERGS, por meio dos editais 08/2023 e 09/2023.

Referências

Amin, M. et. al (2022). Android malware detection through generative adversarial networks. *TETT*, 33(2).

Brown, A., Gupta, M., and Abdelsalam, M. (2024). Automated machine learning for deep learning based malware detection. *Computers & Security*, 137:103582.

Canbek, G., Taskaya Temizel, T., and Sagiroglu, S. (2021). BenchMetrics: A systematic benchmarking method for binary classification performance metrics. *NCA*, 33(21).

Casola, K. et. al. (2023). DroidAugmentor: uma ferramenta de treinamento e avaliação de cGANs para geração de dados sintéticos. In *SBSeg*.

Choi, E. et. al. (2017). Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pages 286–305.

Esteban, C., Hyland, S. L., and Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv preprint arXiv:1706.02633*.

Kouliaridis, V. and Kambourakis, G. (2021). A comprehensive survey on machine learning techniques for Android malware detection. *Information*, 12(5):185.

Li, J., He, J., Li, W., Fang, W., Yang, G., and Li, T. (2024). SynDroid: An adaptive enhanced Android malware classification method based on CTGAN-SVM. *Computers & Security*, 137:103604.

Mimura, M. (2020). Using fake text vectors to improve the sensitivity of minority class for macro malware detection. *JISA*, 54:102600.

Nogueira, A. et. al. (2024). MalSynGen. `https://github.com/SBSegSF24/MalSynGen`.

Park, N. et. al (2018). Data synthesis based on Generative Adversarial Networks. *arXiv preprint arXiv:1806.03384*.

Paullada, A. et. al. (2021). Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11).

Platzer, M. and Reutterer, T. (2021). Holdout-Based Empirical Assessment of Mixed-Type Synthetic Data. *Frontier in Big Data*.

Rainio, O., Teuho, J., and Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1):6086.

Rajabi, A. and Garibay, O. O. (2022). TabfairGAN: : Fair Tabular Data Generation with Generative Adversarial Networks. *ML and Knowledge Extraction*, 4(2):488.

Rocha V. et. al (2023). AMGenerator e AMExplorer: Geração de metadados e construção de datasets android. In *Anais Estendidos do XXIII SBSeg*. SBC.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. biom. bull., 1, 80.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling Tabular Data Using Conditional GAN. *Advances in NIPS*, 32.

Xu, L. and Veeramachaneni, K. (2018). Synthesizing Tabular Data Using Generative Adversarial Networks. *arXiv preprint arXiv:1811.11264*.