Reproducibility – Hands On [Assignment #1]

Cristiano Nicolau
Mineração de Dados em Larga Escala 24/25
Departamento de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro
Aveiro, Portugal
cristianonicolau@ua.pt

***Abstract*—A reprodutibilidade é um pilar essencial para a ciência, especialemnte em áreas como machine learning. Estudos indicam que a replicação de resultados frequentemente encontra obstáculos e não é possivel replicar os resultados. Este trabalho tem como objetivo a replicação da metodologia proposta por Borah et al. para a construção de modelos de detecção de malware. Durante o processo de replicação, diversas limitações foram encontradas, como a falta de acesso ao dataset original, ausência de informações detalhadas sobre o de pré-processamento e a indefinição dos hiperparâmetros usados. Apesar destes problemas, estas limitações foram superadas, dividindo os dados em treino e teste, balanceando as classes e otimizando os hyper parametros. Embora os resultados obtidos a partir dessas suposições sejam identicos aos do estudo original, a falta de informações no artigo impediu uma replicação precisa e rigorosa, levantando preocupações sobre a transparência e a clareza na documentação deste estudo.**

I. Introdução

Como escrito no artigo[1], o uso de modelos de machine learning (ML), tem-se tornado cada vez mais comum no nosso dia a dia em diversas aplicaçoes, desde modelos de classificação a modelos de previsão. O número de estudos científicos que utilizam modelos de ML tem aumentado significativamente, usando modelos de previsão e classificação em diversas areas como a saude, o ambiente, a politica, entre outros. Mas isto tambem leva a problemas, uma grande parte destes estudos revelam falhas na reprodutibilidade desses modelos, muito deles porque não tem informaçao em relaçao ao codigo e aos modelos, outros porque nao explicam nos artigos como modelos foram criados e/ou usados.
Para avaliar o impacto destas falhas, neste relatorio vou replicar a metodologia e os modelos usados por Borah[2] na deteção de malware ou goodware em aplicações Android.

II. Metodologia

Seguindo a metodologia presente no artigo[2], existem dois passos importantes, a aquisição do dataset o pre-processamento, e o desenvolvimento dos modelos de classificacão.

A. Dataset

No artigo analisado são trabalhados dois conjuntos de dados diferentes: TUMALWD e TUANDROMD, mas no artigo nao disponibilizam qualquer acesso quer ao codigo usado quer aos conjuntos de dados usados, revelando-se uma barreira na reprodutibilidade. Devido a isto não foi possivel localizar o conjunto de dados TUMALWD e o conjunto TUANDROMD foi encontrado[3], apesar de não ser possivel fazer uma confirmação que este é o conjunto de dados usado no artigo.

1) Comparação dos Datasets: Na secçao 4.D do artigo, fazem uma analise das caracteristicas do conjunto de dados TUANDROMD, podendo assim fazer uma comparação entre o nosso conjunto de dados e o conjunto de dados usado pelo artigo[3].

Table I: Comparação entre o Dataset descrito no Artigo E O Dataset usado

Dataset	**Valores**				**Features**	
	Total	GoodWare	Malware	Labels	APIs	Permissões
Artigo	25 553	1000	24 553	72	186	178
Dataset	4464	899	3565	2	27	214

Como podemos ver na Tabela I, os datasets possuem diferenças significativas em termos de tamanho e características, o nosso dataset é muito mais limitado em relação ao descrito no artigo, devido a esta diferença tão significativa fica muito mais dificil a replicação do estudo feito por parte dos autores do artigo. Apesar disto, os autores indicam as melhores 15 features do dataset (provavelmente usadas na aplicação dos modelos), faltando uma destas features no nosso dataset *'Landroid/location/LocationManager; -getLastKnownLocation'*, o que leva a crer que apesar das diferenças no dataset os resultados podem ter alguma semelhança.

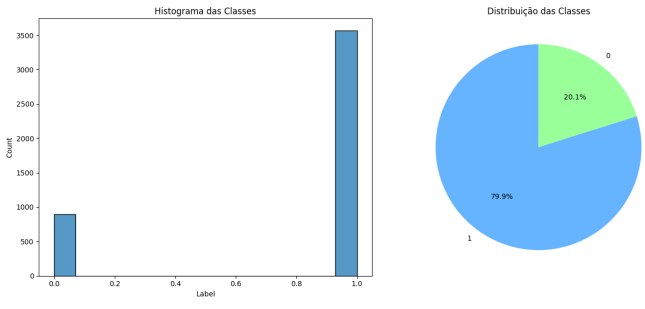

Figure 1: Balanceamento do Dataset

2) Pré Processamento: Mais uma vez existem alguns problemas, que limitam a reprodutibilidade do artigo. Os autores falam do desbalanceamento das classes, onde dizem que o dataset nao está *'perfeitamente balanceado'*, alem disso nao dizem se separaram os dados em treino e teste. Devido a

Table II: Comparação de resultados entre os modelos treinados com todas as features e as 15 principais features.

Modelo	Full Features				Top15 Features			
	Accuracy	F1 Score	Precision	Recall	Accuracy	F1 Score	Precision	Recall
Random Forest	99.6	99.6	99.5	99.7	99.2	99.2	99.6	98.7
Extra Tree	99.7	99.7	99.7	99.7	99.3	99.2	99.8	98.7
Ada Boost	98.5	98.5	99.0	98.0	98.3	98.3	98.5	98.0
XG Boost	99.5	99.5	99.4	99.6	99.1	99.0	99.4	98.7
Gradient Boosting	99.5	99.5	99.3	99.7	99.1	99.1	99.5	98.8

Nota: Os valores apresentados são a média ponderada de cada métrica e foram obtidos após K-Fold Cross Validation com cv = 10 e hyperparameter tuning para cada modelo. Os testes foram realizados com todas as features disponíveis e com as 15 principais features identificadas no artigo base.

isso, como existe um grande desbalanceamento das classes (1 - malware, 0 - goodware) como podemos ver na Figura1, fiz um balanceamento das classes atraves de um random oversampling da classe com menos instancias (goodware) e fiz uma devisão do dataset de 80% para treino e 20% para teste.

B. Modelos

O artigo usa 5 modelos de classificação : Random Forest[4], Extra Tree[5], Ada Boost[6], Xg Boost[7] e Gradient Boosting[8]. Apesar que, o artigo nao meciona nunca os hyperpa- rameters usados, ou se fazem hyperparameter tunning. Além disso no artigo mencionam as 15 melhores features, mas não mencionam nunca se usaram apenas essas 15 features para treinar os modelos ou todas as features. Referem apenas o uso de 10 folds para Cross Validation. Isto mostra mais uma vez a grande dificuldade na replicação deste estudo. Devido a isso, a metodologia usada foi treinar os modelos com os seus hyperparametes deafult, após isso fiz a otimização dos hyperparameters através do GridSearchCV[9] e no fim fiz Cross Fold Validation[10] com 10 folds. Primeiramente treinei os modelos com todas as features e após isso apenas com as 15 melhores features referidas pelo autor.

C. Resultados

Após o desenvolvimento dos 5 modelos de classificação, podemos aplica-los ao dataset completo ou as 15 melhores features indicadas pelos autores do artigo. Após isso podemos comparar os resultados entre os varios modelos do artigo e os modelos feitos por mim. A tabela II representa os resultados de todas as metricas após a aplicação de Cross Validation e Hyperparameter tunning.

Table III: Comparação de resultados entre os modelos treinados com todas as features, as 15 principais features e o dataset TUANDROMD.

Modelo	Test Set		Artigo
	Full Features	Top 15 Features	Accuracy
Random Forest	99.6	99.2	98.7
Extra Tree	99.7	99.3	98.8
Ada Boost	98.5	98.3	97.9
XG Boost	99.5	99.1	97.8
Gradient Boosting	99.5	99.1	97.4

Na tabela III podemos comparar a accuracy entre os modelos do artigo e os modelos aplicados a todo o meu dataset e as top 15 features apresentadas no artigo. Como é possivel ver, tive melhores resultados nos modelos aplicados a todo o dataset do que as 15 melhores features defenidas pelos autores. De acordo com o artigo, os modelos com maior accuracy foram o Random Forest e Extra Tree.

III. Conclusão

Concluindo, através da informação partilhada no artigo não é possível replicar os resultados obtidos pelos autores, devido à falta de detalhes acerca dos procedimentos realizados durante o treino, teste e avaliação dos modelos. Além disso, os autores não fornecem acesso aos datasets completos ou a uma explicação de como as variáveis foram selecionadas o que impede a reprodução do estudo. A ausência de toda esta informação torna a replicação dos resultados impossivel, levantando preocupações sobre a transparência da pesquisa. Para garantir a confiabilidade dos resultados e facilitar a replicação dos resultados, seria necessário que mais informações sobre o processo processamento dos dados, criação dos modelos, definição dos hiperparâmetros e avaliação dos modelos fossem compartilhadas, como o código usado, versões de bibliotecas utilizadas e uma descrição completa do processo de Cross Validation e da otimização dos hiperparâmetros. Além disso, seria ainda interessante que os autores também compartilhassem outras métricas de avaliação, como a curva ROC e a learning curve, porque estes gráficos oferecem insights sobre o desempenho e a estabilidade dos modelos ao longo do treino. Além disso, seria ainda interessante incluir outras métricas como recall e precision, que são essenciais para medir a sensibilidade dos modelos em relação à classificação correta dos dados. No meu caso, todas estas métricas estão disponíveis no notebook, o que pode ser útil para a comparação e análise do comportamento dos diferentes modelos.Mesmo assim, é possível observar algumas semelhanças entre a accuracy dos modelos do artigo e dos meus modelos. Contudo, essas comparações podem ser limitadas devido à falta de detalhes sobre a metodologia exata utilizada pelos autores.

Por fim, é importante que estes estudos realizados tenham uma maior transparência e façam uma melhor descrição da metodologia usada no desenvolvimento de modelos de machine learning de forma a garantir reprodutibilidade dos resultados e das metodologias, de forma a haver avanços científicos.

REFERENCES

[1] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in ml-based science," 2022.

[2] P. Borah, D. Bhattacharyya, and J. Kalita, "Malware dataset generation and evaluation," in *2020 IEEE 4th Conference on Information Communication Technology (CICT)*, 2020, pp. 1–6.

[3] P. Borah and D. K. Bhattacharyya, "Tuandromd (tezpur university android malware dataset)," https://doi.org/10.24432/C5560H, 2020, uCI Machine Learning Repository.

[4] "Randomforestclassifier — scikit-learn documentation," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[5] "Extratreesclassifier — scikit-learn documentation," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[6] "Adaboostclassifier — scikit-learn documentation," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[7] "Xgboost documentation," 2024. [Online]. Available: https://xgboost.readthedocs.io/en/stable/

[8] "Gradientboostingclassifier — scikit-learn documentation," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

[9] "Grid search — scikit-learn documentation," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html

[10] "Cross-validation: evaluating estimator performance," 2024. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html