Reproducibility - Malware Dataset Generation and Evaluation

1st Mia Krsticevic
Department of Electronics, Telecommunications and Informatics
University of Aveiro
Aveiro, Portugal
mkrsticevic@ua.pt
mk54058@fer.hr

***Abstract*—This article addresses the challenges of trying to reproduce a classification AI model. The results are reproduced from an article, titled *Malware Dataset Generation and Evaluation*[1], which investigates methods for improving detection of malware. Additionally, this paper provides a critical analysis of the referenced article, highlighting areas where the methodology could have been improved or clarified for better reproducibility.**

Keywords—malware detection, dataset reproducibility, machine learning, classification models, cybersecurity

I. Introduction

Malware has become a common issue in today's digital world, with scams and cyber-attacks happening more frequently every day. It is crucial to try and stop as many threats as possible, even when some system components may initially appear invulnerable. The article *Malware Dataset Generation and Evaluation* explores different methods for better detection of malicious content on two different platforms – iOS and Android. It describes the process of gathering datasets for each platform, the challenges faced during data collection, and the training of machine learning models. Several models were trained to determine which one gives the best results. The following section will provide an overview of the methodology used in the referenced article.

II. METHODOLOGY

The referenced article firstly goes through explanation of the concept of malware. It discusses how the internet is exposed to cyber-attacks, how information is stolen and how the systems are compromised. Two datasets were used in the article, one for each platform. However, due to issues with finding the iOS dataset, the focus is placed only on the Android dataset - *TUANDROM.csv*.

The article further looks into how malware behaves and spreads across the internet and victims' devices. It explains the process of identifying and confirming the malicious content in detail. A key component of the article is the collection of data through an automated Honeynet system, which is designed to capture and analyze potentially harmful files. Once the data is collected, it goes through data analysis and feature extraction within the Cuckoo Sandbox environment where both benign and malicious files are detected and categorized. The dataset is then preprocessed by removing empty data.

Upon reviewing this article, it becomes evident that the dataset preparation and training process were not adequately detailed. This lack of explanation significantly impacts reproducibility, making the process longer and more complex. Providing a complete description of data preprocessing and model training is crucial in a study on classification models. This approach weakens the overall quality of the article.

The article states that the dataset consists of 72 labels and 25,553 instances. However, when retrieving the dataset from the referenced sources and online repositories, 242 labels with 4,465 instances were found instead. This makes the pre-processing journey additionally bigger and more complicated. Also, it is said that there is a big class imbalance, which was confirmed during the reproducibility process. In the article, there is only 1000 instances (3,9 % of the dataset) belonging to the benign class whereas when trying to reproduce, there is 899 instances (20,1 %).

Furthermore, the only steps provided is the use of K-Fold cross validation with the value of K=10, but there is no elaboration on the choice or justification for this parameter. The models applied in the study include Random Forest, Extra Trees, AdaBoost, XGBoost, and Gradient Boosting. However, further details on hyperparameter tuning, training duration and size are not provided.

III. Challenges Encountered

When attempting to reproduce the results presented in the article, one of the primary challenges was handling inconsistencies in data frame shapes. Missing values were removed and constant and monotonic columns were dropped, since they do not contribute to the model's performance. During the pre-process, it is important to encode categorical labels and scale numeric labels if necessary. *LabelEncoder* function was used for encoding binary output label (0 for benign, 1 for malware).

Because the article indicated a significant class imbalance, it was important to check and reduce the imbalance, to improve model's generalization. To achieve this, *undersampling* was done—a technique that randomly removes instances from the majority class to make the class proportions more balanced [2].

Additionally, the dataset contained an excessive number of labels (200) even after dropping constant columns, indicating dimensionality curse. Not all of the labels will give valuable information for our model. Principal Component Analysis

(PCA) is used for dimensionality reduction. Prior to PCA, *StandardScaler* function was used to normalize values to the [0,1] range, ensuring that outliers do not disproportionately influence the model's performance. PCA was applied with n_components=0.95, retaining 95% of the variance in the data. As a result, the number of components was reduced to 68. The dataset was split into train (80%) and test (20%) set using the *train_test_split* function. K-Fold cross-validation (K=10) was implemented, as specified in the article. No additional hyperparameters were added in the training process.

The results were somewhat similar to those reported in the article, but slightly improved, making the reproducibility process successful.

IV. Suggestions for Improvement

The model's performance could potentially be improved with more data. Specifically, collecting benign data may enhance the overall performance. However, one of the major issues with the article is that the entire preprocessing, training, and evaluation process is poorly described, which makes it difficult to replicate and assess the methodology effectively.

Also, the results are represented only through accuracy, which is also not ideal. Accuracy represents the ratio of correctly predicted instances to the total number of instances [3]. However, it does not give us good feedback when there is class imbalance. In cases of severe class imbalance, a model may appear to perform well because it predominantly predicts the majority class. But this just may be simply because the model is biased towards the majority class, leading to poor generalization and ineffective predictions for the minority class. Additionally, accuracy does not distinguish between false positive and false negative instances. Therefore, it is important to consider other metrics, such as precision and recall, or the F1 score, which balances both.

Moreover, in terms of handling class imbalance, alternative approach to *undersampling* could be oversampling (to increase the minority class). However, I believe *undersampling* is a more appropriate option in this context, as it retains the true instances rather than relying on synthetic data generation.

Regarding the choice of models, I believe the selected ones were appropriate for the task. If more complex models were chosen, such as Deep Learning models they would likely be too complex for this dataset.

V. Conclusion

All in all, with a few changes, the referenced article could achieve a higher level of quality. It is essential to describe the entire process in detail to ensure that readers fully understand the methodology and can successfully reproduce the results. While the article primarily focuses on dataset generation, as its title suggests, the evaluation and model training sections do not meet the expected level of depth. The final part appears somewhat rushed, which weakens the overall analysis. However, the explanation of malware itself is well-structured and clearly presented. This paper builds upon the referenced work by addressing the whole training and evaluation process. By bridging these gaps, it offers a more complete understanding of the problem and the applied methodology.

[1] P. Borah, D. Bhattacharyya, and J. Kalita, "Malware Dataset Generation and Evaluation," *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, Chennai, India, 2020, pp. 1-6, doi: 10.1109/CICT51604.2020.9312053.

[2] D. Santiago, "Balancing imbalanced data: Undersampling and oversampling techniques in Python," *Medium*, Jun. 5, 2023. [Online]. Available:https://medium.com/@daniele.santiago/balancing-imbalanced-data-undersampling-and-oversampling-techniques-in-python-7c5378282290. Accessed: Mar. 7, 2025.

[3] J. Šnajder, Machine Learning 1, Model evaluation, version 1.3, Univ. of Zagreb, Faculty of Electrical Engineering and Computing, academic year 2022/2023. Available to students at FER.